

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ZÁZNAMNÍK KLIMATICKÝCH VELIČIN

DATALOGGER OF ENVIRONMENTAL VALUES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Raichl

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Petr Raichl

ID: 182993

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Záznamník klimatických veličin

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a zrealizujte zapisovač klimatických dat se solárním napájením.

1. Proveďte literární a internetovou rešerši.
2. Navrhněte koncept záznamníku a zpracujte energetickou bilanci.
3. Navrhněte elektroniku solárně napájeného záznamníku, který umožní dlouhodobé zaznamenání měření teploty, tlaku, vlhkosti a případně dalších klimatických veličin. Systém navrhněte a zrealizujte jako miniaturní, mikrokontrolérem řízený systém, vybavený solárním napájením.
4. Navrhněte komplexní programové vybavení pro nastavení záznamníku a zpracování naměřených údajů.
5. Výsledný systém zhodnoťte.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá konstrukcí záznamníku klimatických dat se solárním napájením. Práce začíná průzkumem trhu. Následně práce pokračuje tvorbou konceptu vlastního záznamníku. Poté je navržena elektronika záznamníku. Záznamník je postaven jako mikrokontrolérem řízený systém, který čte data ze snímačů, zapisuje data na paměťovou kartu a odesílá data pomocí LoRa technologie do cloudu Českých radiokomunikací. Následně jsou data přeposílána do MySQL databáze. Program pro mikrokontrolér je vytvořen v jazyce c. Konfigurace mikrokontroléru se provádí přes rozhraní USB pomocí PC aplikace vytvořené v C#. Pro vizualizaci naměřených dat byla vytvořena webová databázová aplikace v ASP.NET Core.

KLÍČOVÁ SLOVA

Záznamník, solární panel, klimatická data, snímač, mikrokontrolér, STM32L151, LoRa, SD karta, C#, MVVM, WPF, MySQL, MVC, ASP.NET Core, Amazon Web services

ABSTRACT

The thesis is aimed on construction of solar panel powered datalogger of environmental values. The thesis begins by research of commercial dataloggers. Then the concept of datalogger was created. Then the electronics of datalogger was designed. Datalogger is built as a system driven by microcontroller. The microcontroller reads data from sensors, writes data to memory card and sends data to Czech radiocommunications cloud by LoRa technology. Then the data are resend to MySQL database. Program for microcontroller was made in C language. Configuration of datalogger is done through USB interface by PC application which was created in C# language. For visualisation of measured data ASP.NET Core application was created.

KEYWORDS

Datalogger, solar panel, climate data, sensor, microcontroller, STM32L151, LoRa, SD card, C#, MVVM, WPF, MySQL, MVC, ASP.NET Core, Amazon Web Services

RAICHL, Petr. *Záznamník klimatických veličin*. Brno, 2020, 107 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Záznamník klimatických veličin“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 22.5.2020

.....
podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc.Ing. Zdeňku Bradáčovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno 22.5.2020

.....

podpis autora

Obsah

Úvod	13
1 Průzkum trhu	14
1.1 Průzkum trhu	14
2 Návrh konceptu záznamníku	16
2.1 Snímače	17
2.2 Paměťová SD karta	20
2.3 USB	22
2.4 Internet věcí	23
2.4.1 Sigfox	23
2.4.2 LoRa	24
2.5 Mikrokontrolér	27
2.6 Energetická bilance	30
3 Návrh HW	31
3.1 Návrh hlavní desky	31
3.1.1 Schéma zapojení IOT modulu RN2483	32
3.1.2 Schéma zapojení SD karty	32
3.1.3 USB komunikace	32
3.1.4 Mikrokontrolérová část	33
3.1.5 Schéma napájecí části	34
3.2 Návrh zdroje	35
3.2.1 Koncept 1 - superkapacitor	35
3.2.2 Měření solárního článku	37
3.2.3 Koncept 2 - baterie i superkapacitor	39
3.2.4 Koncept 3 - Lithium-Titanová baterie	41
3.2.5 Srovnání konceptů zdroje	42
3.2.6 Obvody pro dobíjení baterie	43
4 Sestavení a oživení zařízení	46
4.1 LTC3105	46
4.2 LTC3129	46
4.3 Hlavní deska	47
4.3.1 Napájecí část	47
4.3.2 Mikrokontrolér	48
4.3.3 IOT modul RN2483	48
4.3.4 Snímače	49

4.3.5	Paměťová karta	49
4.3.6	FTDI pro komunikaci s PC	50
4.3.7	Nová verze hlavní desky	50
4.3.8	Krabička	51
5	Programové řešení	52
5.1	Program mikrokontroléru	52
5.1.1	Hlavní smyčka programu	52
5.1.2	Časované přerušení	54
5.1.3	Obsluha snímačů	55
5.1.4	Obsluha IOT modulu RN2483	55
5.1.5	Obsluha komunikace s PC	57
5.1.6	Komunikační protokol pro komunikaci s PC	57
5.1.7	Ukládání dat na SD kartu	60
5.1.8	Optimalizace na nízký odběr energie	62
5.2	Program pro ovládání zařízení z PC	64
5.2.1	Návrhový vzor MVVM	65
5.2.2	Model - Obsluha COM Portu	65
5.2.3	ViewModel	66
5.2.4	Views	66
5.3	Návod na konfiguraci zařízení z PC	69
5.4	Vizualizace měřených dat	70
5.4.1	Služby CRA	70
5.4.2	AWS	71
5.4.3	Databáze	72
5.5	Tvorba webové aplikace	73
5.5.1	Požadavky na aplikaci	74
5.5.2	ASP.NET Core	74
5.5.3	Návrhový vzor MVC	74
5.5.4	Model	76
5.5.5	Controllers	77
5.5.6	Views	78
5.5.7	Publikování aplikace	82
6	Závěr	83
6.1	Koncept záznamníku	83
6.2	HW	84
6.3	SW	84
6.3.1	SW pro mikrokontrolér	84

6.3.2	PC aplikace	85
6.3.3	Vizualizace naměřených dat	85
6.3.4	Webová aplikace	85
6.4	Zhodnocení	85
6.4.1	Návrhy do budoucna	86
Literatura		87
Seznam příloh		90
A Schéma zařízení		91
A.1	Schéma hlavní desky	91
A.2	Schéma LTC-3105	91
A.3	Schéma LTC-3129	91
B Desky plošných spojů		95
B.1	DPS hlavní desky	95
B.1.1	Horní strana	95
B.1.2	Spodní strana	95
B.1.3	Osazovací schéma	95
B.2	DPS LTC-3105	99
B.2.1	DPS	99
B.2.2	Osazovací schéma	99
B.3	DPS LTC-3129	102
B.3.1	DPS	102
B.3.2	Osazovací schéma	102
C Seznam součástek		105
C.1	Hlavní deska	105
C.2	LTC-3105	106
C.3	LTC-3129	106
D Obsah přiloženého CD		107

Seznam obrázků

1.1	Ukázka několika záznamníků klimatických dat[1, 2, 3]	15
2.1	Zjednodušené blokové schéma záznamníku	17
2.2	Snímač teploty a vlhkosti SHT-85[4]	18
2.3	Schéma zapojení SHT-85 s nadřazeným mikrokontrolérem[4]	18
2.4	Závislosti odchylek teploty snímače SHT-85[4]	19
2.5	Závislosti odchylek vlhkosti snímače SHT-85[4]	19
2.6	Snímač tlaku BMP280[5]	20
2.7	Velikosti pamětových karet[6]	20
2.8	Vnitřek SD karty[6]	21
2.9	Struktura rámce s příkazem[6]	22
2.10	Příklad odezev[6]	22
2.11	FT232 - komunikace s PC[8]	23
2.12	IOT - Technologie v závislosti na dosahu a odběru energie[9]	24
2.13	Vysílací pásma sigfox[12]	24
2.14	Průběh odběru proudu - sigfox[12]	25
2.15	ISO/OSI - LoRa[9]	25
2.16	Topologie LoRaWan sítě[9]	26
2.17	Design LoRa zařízení[9]	27
2.18	Struktura RN2483[10]	28
2.19	Blokové schéma záznamníku doplněné o sběrnice	29
3.1	Blokové schéma hlavní desky	31
3.2	Schéma zapojení IOT modulu RN2483	32
3.3	Schéma zapojení SD karty	33
3.4	Schéma zapojení části určené pro USB komunikaci	34
3.5	Schéma mikrokontrolérové části	35
3.6	Schéma napájecí části	36
3.7	Blokové schéma zdroje	36
3.8	Simulační schéma	38
3.9	Průběh napětí na kondenzátoru	38
3.10	Schéma měření VA charakteristiky solárního článku	39
3.11	VA char. solárního článku - jasno(nahoře), zataženo(dole)	40
3.12	Blokové schéma nového konceptu zdroje	40
3.13	Vylepšené blokové schéma hlavní desky	41
3.14	Závislost kapacity baterie na teplotě - vlevo Lithium, vpravo Lithium-titan [13]	42
3.15	Schéma zapojení s obvodem LTC3105	44
3.16	Schéma zapojení s obvodem LTC3129	45

4.1	Osazená deska plošného spoje s obvodem LTC3105	46
4.2	Osazená deska plošného spoje s obvodem LTC3129	47
4.3	Osazení hlavní desky s náhradním mikrokontrolérem na univerzální dps	49
4.4	Osazená 2. verze dps hlavní desky	50
4.5	Zařízení umístěné v krabici	51
5.1	Technika Foreground/Background Task[18]	52
5.2	Vývojový diagram hlavní smyčky programu	53
5.3	Vývojový diagram časovaného přerušení	54
5.4	Vývojový diagram obsluhy snímačů	55
5.5	Vývojový diagram obsluhující vysílání dat	56
5.6	Vývojový diagram obsluhující příjem dat z modulu RN2483	57
5.7	Vývojový diagram obsluhy příjmu dat od RN2483 na uart	58
5.8	Vývojový diagram obsluhy komunikace s PC	59
5.9	Struktura zprávy od PC směrem k slave zařízení	59
5.10	Struktura zprávy od slave zařízení směrem k PC	60
5.11	Struktura zprávy typu Nastav datum a čas	60
5.12	Vývojový diagram pro obsluhu uart od obvodu ftdi	61
5.13	Ukázka textového souboru se záznamem dat	61
5.14	Diagram tříd StmService(vlevo) a MainPageViewModel(vpravo)	67
5.15	Diagram tříd náležících ViewModels	68
5.16	Ukázka hlavní stránky aplikace	68
5.17	Diagram třídy BooleanToBrushConverter	68
5.18	Ukázka stránky pro nastavení parametrů komunikace	69
5.19	Ukázka CRA IOT portálu	70
5.20	Tok dat od zařízení k aplikaci pro vizualizaci [20]	71
5.21	Ukázka zprávy formátu JSON [20]	72
5.22	Amazon Serverless architektura [21]	72
5.23	Diagram MVC [22]	75
5.24	Diagramy tříd modelů	76
5.25	Diagramy třídy HomeController	77
5.26	Diagramy třídy DeviceController	78
5.27	Diagramy třídy UserController	78
5.28	Ukázka hlavní stránky aplikace	79
5.29	Ukázka seznamu zařízení náležících uživateli	79
5.30	Ukázka View pro vytvoření nového zařízení	80
5.31	Ukázka View pro smazání zařízení	80
5.32	Ukázka grafu teploty	80
5.33	Ukázka domovské obrazovky uživatele s rolí Admin	81
5.34	Ukázka View se seznamem zaregistrovaných uživatelů	81

5.35	Ukázka registrace nového uživatele	82
6.1	Finální koncept záznamníku i s hlavními obvody	83

Seznam tabulek

1.1	Průzkum trhu	14
2.1	Základní parametry vybraných snímačů	20
2.2	Základní parametry mikrokontroléru STM32L151	29
2.3	Energetická spotřeba jednotlivých operací	30

Úvod

Tato práce se věnuje oblasti záznamu klimatických dat, a to především v podobě návrhu a tvorby funkčního prototypu zařízení, které je schopno sbírat klimatická data, ukládat tato data a takto pracovat dlouhodobě, neboť zařízení bude napájeno ze solárního panelu, tzn. že záznam nebude přerušen vybitím baterie. Záznamník tohoto typu bude moci být umístěn kdekoliv, kde budou vhodné světelné podmínky, tzn. dostatečná intenzita světla dopadající na solární panel.

Přestože zařízení bude napájeno ze solárního panelu, bude třeba zařízení optimalizovat pro nízký odběr energie. Solární panel nebude sloužit jako zdroj energie přímo pro zařízení, ale bude využit pro dobíjení akumulárního prvku, tzn. baterie nebo superkapacitoru, které se dnes vyrábí s kapacitou v hodnotách i stovek Faradů.

Snímače klimatických dat, tedy snímače teploty, vlhkosti a tlaku bude třeba vybrat vzhledem k rozsahu měřených hodnot a podmínkám venkovního prostředí. Dále na základě kompromisu mezi přesností, odběrem energie a velikostí.

Hlavní jednotkou zařízení bude mikrokontrolér, který bude mít na starosti sběr a uložení dat ze snímačů. Dále bude komunikovat s vnějším světem, a to minimálně ve dvou případech: V případě konfigurace záznamu dat a v případě poskytování dat. Pro konfiguraci zařízení se nabízí sériová sběrnice USB, jakožto v současnosti hojně používaný způsob připojení zařízení k PC. Vzhledem ke skutečnosti, že zařízení bude konfigurováno z PC přes sběrnici USB, bude potřeba vytvořit program pro PC, který tuto konfiguraci bude provádět. Pro případ poskytování dat bude využito paměťové SD karty, a zároveň některého z bezdrátových přenosů dnes moderních IOT - Internet Of Things. Výběr technologie IOT bude diskutován později v této práci. Následně na základě vybrané IOT technologie bude potřeba navrhnout řešení pro zobrazení, vyhodnocení naměřených dat.

1 Průzkum trhu

V této části práce je popsána aktuální nabídka na trhu se záznamníky klimatických veličin. Na základě průzkumu trhu se bude následně vytvářet koncept záznamníku navrhovaného v této práci.

1.1 Průzkum trhu

Při průzkumu trhu byly využity především portály cz.Farnell.com, tme.eu a cz.rs-online.com. Většina záznamníků nabízí záznam teploty a vlhkosti při napájení z baterie. Na základě průzkumu lze tvrdit, že většina výrobků využívá sériovou sběrnici USB pro konfiguraci zařízení i pro transport dat ze zařízení do PC, kdy většina výrobců nabízí software pro grafické zobrazení naměřených dat. Mnohdy mohou tyto výrobky být i trvale připojeny pomocí USB a být tímto způsobem napájeny. V mnohem menší míře se vyskytují výrobky, které posílají naměřená data po síti pomocí WiFi technologie. U těchto výrobků je těžké říci jak dlouho vydrží provoz na baterie, neboť to závisí na několika faktorech jako např. perioda vysílání, provozní teplota.

Většina zařízení nabízí kapacitu paměti několik desítek tisíc až stovek tisíc záznamů. Jen málo zařízení nabízí ukládání dat na paměťovou kartu. Což je překvapivé, vzhledem k možnosti paměťovou kartu s daty jednoduše vytáhnout a vložit do počítače. Samozřejmě za předpokladu implementace nějakého pro PC typického souborového systému, např. fat32.

V tab.1.1 jsou uvedeny základní parametry několika typických představitelů vybraných cenových tříd. Příslušné katalogové listy se nachází v příloze y.

Model	Rozhraní	Log. interval	Snímač teploty	Snímač vlhkosti	Přesnost tep(%)	Přesnost vlh(%)	Cena (Kč)
TGP-4020	USB	1s-10dní	Termistor		0.35		6425
HL-1D	USB	30s-24h	Termistor	Kapacitní	+0.3°C	+3%RH	3240
EL-USB-2-LCD	USB	1s-12h	Neuveden	Neuveden	0.4°C	2.25%RH	2711
PRO-USB-1	USB	1s-12h	Neuveden		+0.2°C		3672
HL-NT2-DP	USB	5s-24h	Pt100	Rotronic HygroClip2	+0.2°C	+1.5%RH	26860

Tab. 1.1: Průzkum trhu

V obr.1.1 jsou zobrazeny ukázky podoby několika záznamníků z tab. 1.1.

Při průzkumu nebyl nalezen žádný výrobek, který by umožňoval napájení ze solárního panelu a tím teoreticky neomezenou dobu sběru dat - samozřejmě s ohledem na kapacitu paměti. Většina výrobků deklaruje dobu životnosti baterie 2 roky.



Obr. 1.1: Ukázka několika záznamníků klimatických dat[1, 2, 3]

2 Návrh konceptu záznamníku

Koncept záznamníku navrhovaného v této práci je navržen na základě zadání a na základě průzkumu trhu. Záznamník by měl být schopen uchovat velké množství naměřených dat - proto se data budou ukládat na paměťovou kartu. Aby se data dala lehce analyzovat po vložení paměťové karty do PC, bude implementován některý z typických souborových systémů.

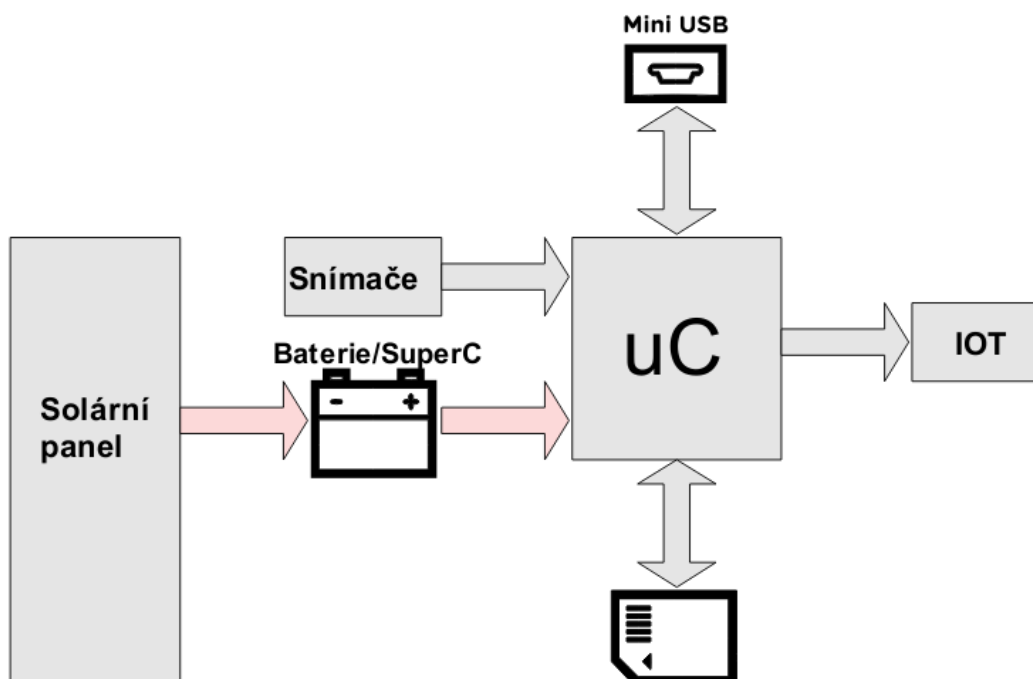
Jak již bylo dříve popsáno, většina záznamníků využívá univerzální sériovou sběrnici USB pro konfiguraci záznamníku a dále pro získání dat ze záznamníku a následné zpracování dat na PC. A proto i záznamník navrhovaný v této práci bude opatřen sběrnici USB.

Pro konfiguraci záznamníku bude napsán program pro PC. Dále bude zařízení podporovat některou z technologií typu IOT - Internet Of Things, což je velká výhoda oproti zařízením popsaným v průzkumu trhu, která bude navrhované zařízení činit modernějším a uživatelé budou mít možnost sledovat data v téměř reálném čase.

Hrubý koncept záznamníku lze tedy shrnout do následujících bodů:

- Napájení - solární panel
- Solární panel dobíjí baterii/superkapacitor
- Hlavní jednotkou je mikrokontrolér
- Nastavení měření se provádí přes USB
- Data ze snímačů se zpracovávají v mikrokontroléru
- Naměřená data se ukládají na paměťovou kartu
- Data lze do počítače přenést vložím paměťové karty do PC
- Podpora některé z technologií IOT a následná vizualizace dat

Na obr. 2.1 je zobrazeno zjednodušené blokové schéma záznamníku.



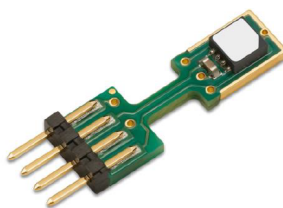
Obr. 2.1: Zjednodušené blokové schéma záznamníku

2.1 Snímače

Vzhledem k požadavku zadání, aby zařízení bylo miniaturní, bylo rozhodnuto, že nebudou použity klasické snímače teploty jako jsou např. platinový snímač, termistor, termočlánek, neboť by bylo potřeba řešit správný návrh měřicího obvodu, popř. kompenzace. Namísto těchto snímačů budou využity snímače typu MEMS, tzn. že jedno pouzdro obsahuje snímač i měřící obvod, který řeší zpracování měřené hodnoty v podobě např. analogově/digitální převod, tovární kalibrace, linearizace, závislost na napájecím napětí a digitální výstup v podobě sběrnice I2C nebo SPI. Díky digitálnímu výstupu v podobě těchto sběrnic nebude třeba řešit analogově/digitální převody.

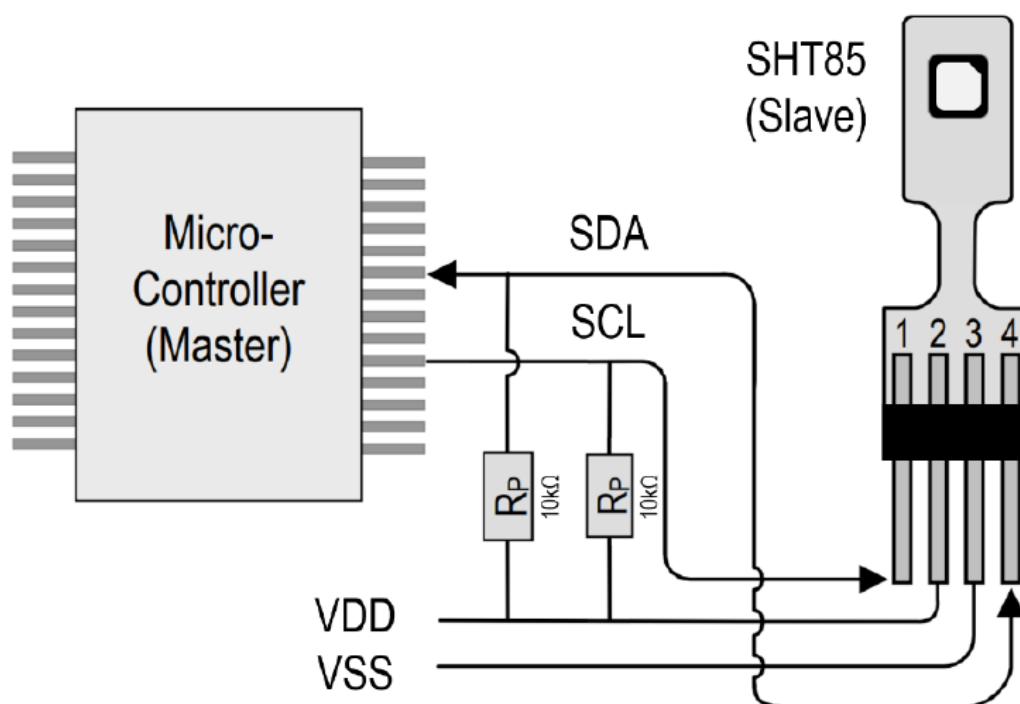
Po dalším průzkumu trhu, tentokrát trhu se snímači, bylo rozhodnuto, že bude využit snímač SHT-85 pro měření teploty a vlhkosti. Tento snímač tedy měří teplotu a vlhkost v jednom pouzdře, přičemž data poskytuje po sběrnici I2C. Tento snímač byl zvolen z několika důvodů:

- V jednom pouzdře obsahuje snímač teploty i vlhkosti
- Přesnost
- Nízký odběr
- Digitální výstup - I2C sběrnice
- Solidně zpracovaný katalogový list



Obr. 2.2: Snímač teploty a vlhkosti SHT-85[4]

Snímač je umístěn na desce plošných spojů, není tedy potřeba jej pájet. Je třeba jej pouze zapojit dle schématu zapojení, viz. obr. 2.3.



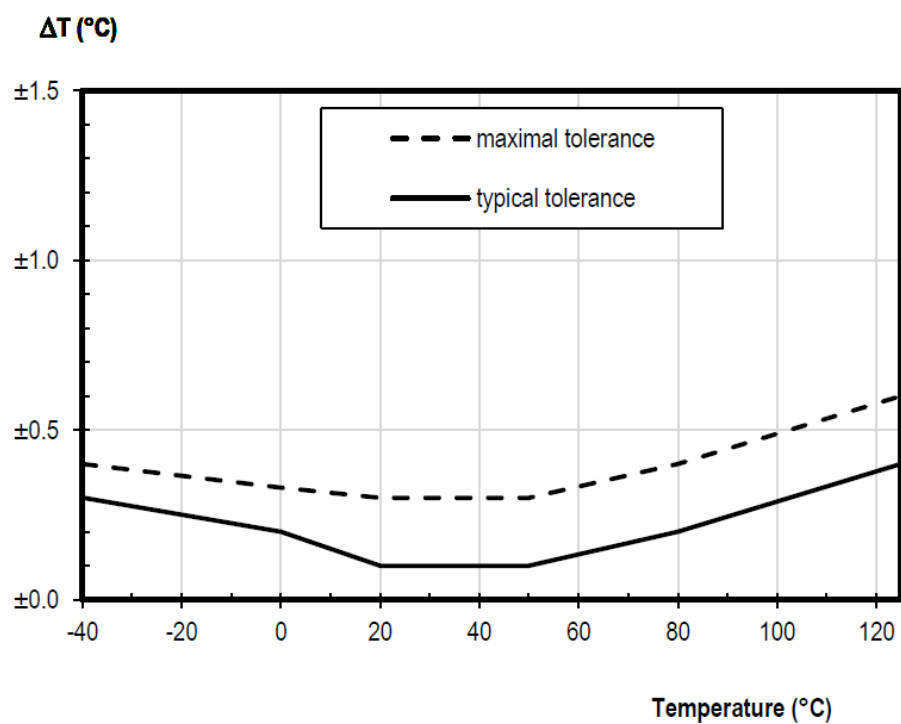
Obr. 2.3: Schéma zapojení SHT-85 s nadřazeným mikrokontrolérem[4]

V tab. 2.1 jsou základní parametry snímače SHT-85.

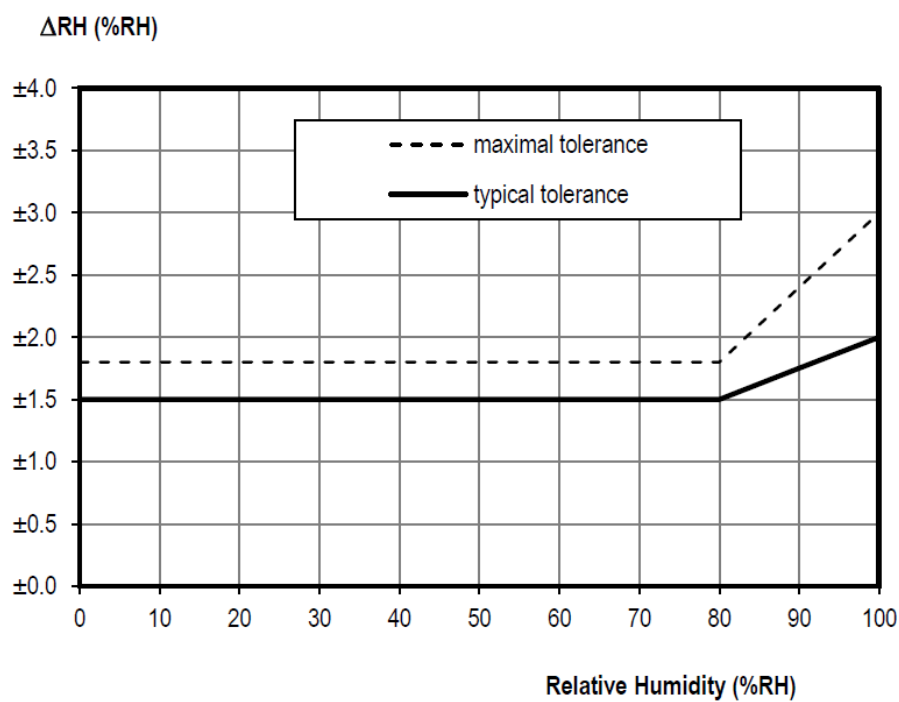
V obr.2.4, 2.5 je zakreslena závislost přesnosti snímače SHT-85 pro teplotu a vlhkost.

Pokud přesnosti snímače udávané v katalogovém listu odpovídají skutečnosti, měl by navržený přístroj být minimálně stejně přesný při měření teploty a vlhkosti jako komerční verze uvedené v tab. 1.1, které cenu záznamníku výrazně převyšují.

Pro měření tlaku bude využit levný snímač BMP280, 2.6, od firmy Bosch. Tento snímač podobně jako snímač SHT-85 disponuje digitálním výstupem sběrnice I2C. Tím tedy máme definováno, jakým způsobem bude mikrokontrolér komunikovat



Obr. 2.4: Závislosti odchylek teploty snímače SHT-85[4]



Obr. 2.5: Závislosti odchylek vlhkosti snímače SHT-85[4]

se snímači - je to sběrnice I2C, která bude popsána později. Základní parametry snímačů jsou uvedeny v tab. 2.1.



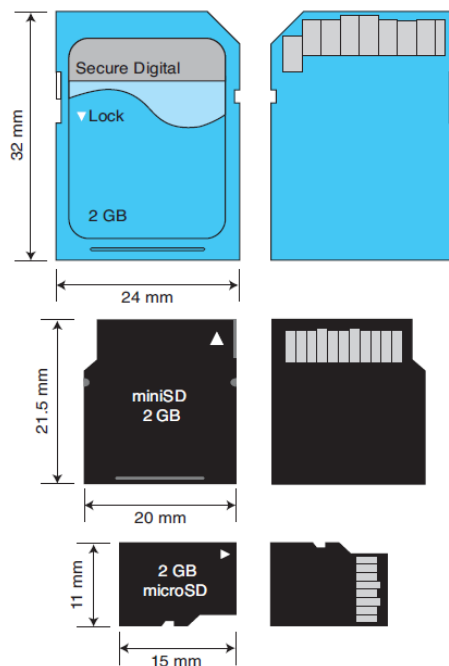
Obr. 2.6: Snímač tlaku BMP280[5]

Snímač	Rozhraní	Nap napětí	Odběr proudu	Provozní teplota	Přesnost tep(%)	Cena (Kč)
SHT85	I2C	2.15-5.5	1.7uA	-40 - 125	0.4°C, 1.8%RH	600
BMP280	I2C	1.71-3.6	2.8uA	-40 - 85	+0.12hPa	100

Tab. 2.1: Základní parametry vybraných snímačů

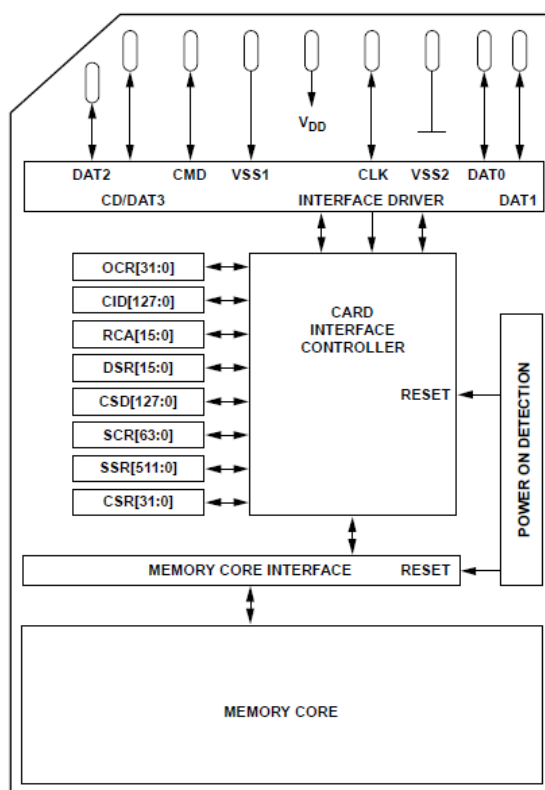
2.2 Paměťová SD karta

SD karta byla představena v roce 1999. Od té doby se stala standardem používaným v přenosných zařízeních jako médium pro uchování např. multimediálních dat v podobě obrázků, hudebních nahrávek, videí, atd. Dnes se vyskytují typicky ve třech velikostech, viz. obr.2.7.



Obr. 2.7: Velikosti paměťových karet[6]

SD karty obsahují rozhraní pinů, paměťové jádro, interní registry a vnitřní kontrolér, viz. obr.2.8. Paměťové jádro je místo pro uložení samotných dat. Interní registry obsahují aktuální informace o kartě. Pinové rozhraní slouží k připojení k mikrokontroléru. Interní kontrolér se stará o zápis, čtení, mazání dat, atd. Nadřazený mikrokontrolér, který ukládá nebo čte data z SD karty se tedy nezajímá o paměťové jádro. Místo toho posílá internímu kontroléru příkazy s daty a interní kontrolér příkazy interpretuje a vykonává.[6]

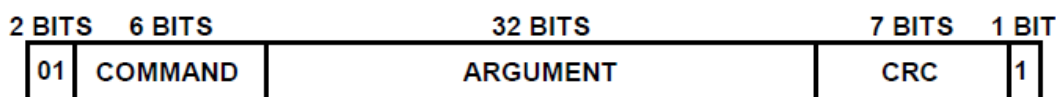


Obr. 2.8: Vnitřek SD karty[6]

SD karty podporují 3 módy komunikace: SPI mód, 1-bit SD mód a 4-bit SD mód. 4-bit SD mód se vyznačuje nejvyšší přenosovou rychlostí, ale zároveň je nejsložitější na implementaci. Nejčastěji používaným módem je SPI mód, který jak už z názvu vyplývá, využívá u mikrokontrolérů často používanou sběrnici SPI.[6] Mikrokontrolér, který bude použit v této práci, bude pravděpodobně obsahovat rozhraní SPI, a proto bude SD karta připojena na toto rozhraní. Proto bude tento SPI mód v této práci popsán.

SPI definuje pouze fyzickou vrstvu, nedefinuje komunikační protokol. Proto se využívá příkazů z módu SD. SD karta v SPI módu může používat jen podmnožinu příkazů používaných v módu SD. Příkazy v módu SD lze najít v [7]. Komunikaci s SD kartou v SPI módu začíná nadřazený mikrokontrolér odesláním rámce s příkazem,

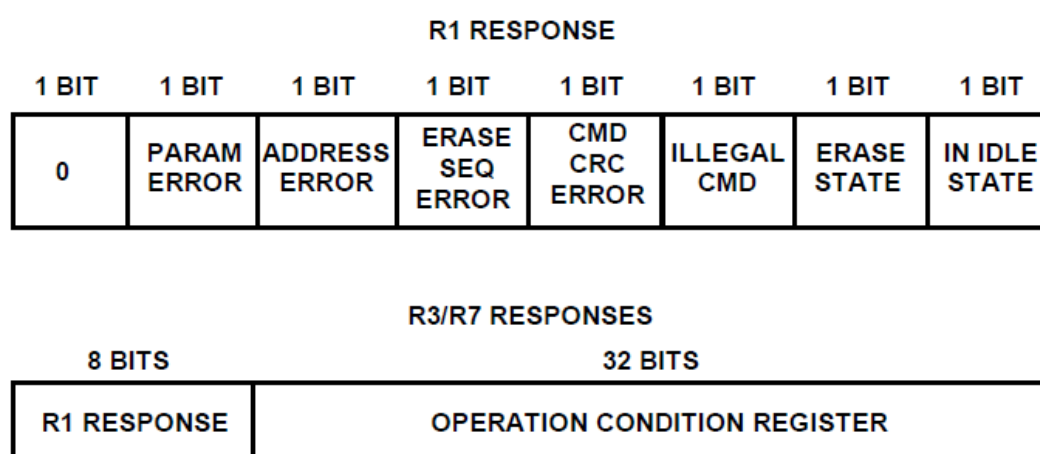
viz. obr.2.9. Karta odpovídá odesláním rámce odezvy nebo chybového rámce.[6]



Obr. 2.9: Struktura rámce s příkazem[6]

Příkazový rámec se skládá ze 6 bytů. Rámec vždy začíná bity 01, které jsou následovány 6-bit číslem příkazu. Následují 4 byty argument ve formátu big endian. Dalších 7 bitů obsahuje CRC kód a poslední bit stop bit.

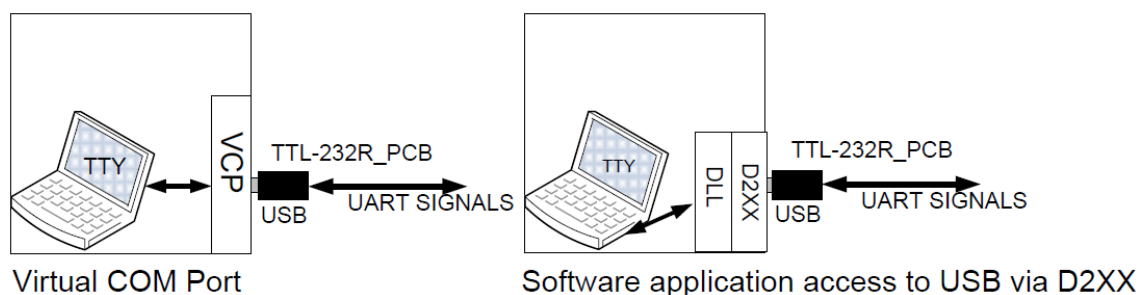
Odezva karty v SPI módu může být typu R1, R3 nebo R7, viz. obr.2.10.



Obr. 2.10: Příklad odezev[6]

2.3 USB

Pro zjednodušení práce s USB bude využit obvod FTDI FT232RL. Obvod FT232RL bude použit jako převodník mezi rozhraním USB na straně osobního počítače a rozhraním UART na straně mikrokontroléru. K obvodu jsou dostupné 2 typy ovladačů pro PC. Díky prvnímu typu se obvod chová jako virtuální sériový port, viz. obr.2.11. Druhý typ umožňuje přímý přístup pomocí dodaných knihoven, viz. obr.2.11. Nespornou výhodou obou řešení je tedy snazší vývoj aplikací pro PC. Převodník může být napájen ze sběrnice USB nebo z externího zdroje.[8] V této práci bude využito napájení ze sběrnice USB z důvodu úspory energie.



Obr. 2.11: FT232 - komunikace s PC[8]

2.4 Internet věcí

Internet věcí je označení sítě fyzických zařízení, které si mohou vyměňovat data prostřednictvím technologií bezdrátového přenosu a internetu. Internet věcí umožňuje zařízením bezdrátově komunikovat a zároveň umožňuje, aby zařízení byla bezdrátově ovládána. Tato zařízení mohou být např. z oblasti zdravotnictví, logistiky, energetiky, metrologie nebo chytrých domácností.[9]

Pro internet věcí se dá vesměs využít kterákoliv bezdrátová technologie přenosu dat:

- WiFi
- Bluetooth
- ZigBee

Tyto technologie jsou vhodné především pro zařízení u kterých není kladen důraz na nízkou spotřebu, typicky zařízení které nejsou napájeny z baterií, ale disponují jiným externím zdrojem napájení.

Pro zařízení u kterých je kladen důraz na nízký odběr energie se používají technologie spadající do kategorie LPWAN-Low Power Wide Area networks. Tyto technologie nabízí velký dosah - až několik desítek km a nízký odběr energie. V současnosti v České republice převažují 2 technologie typu LPWAN:

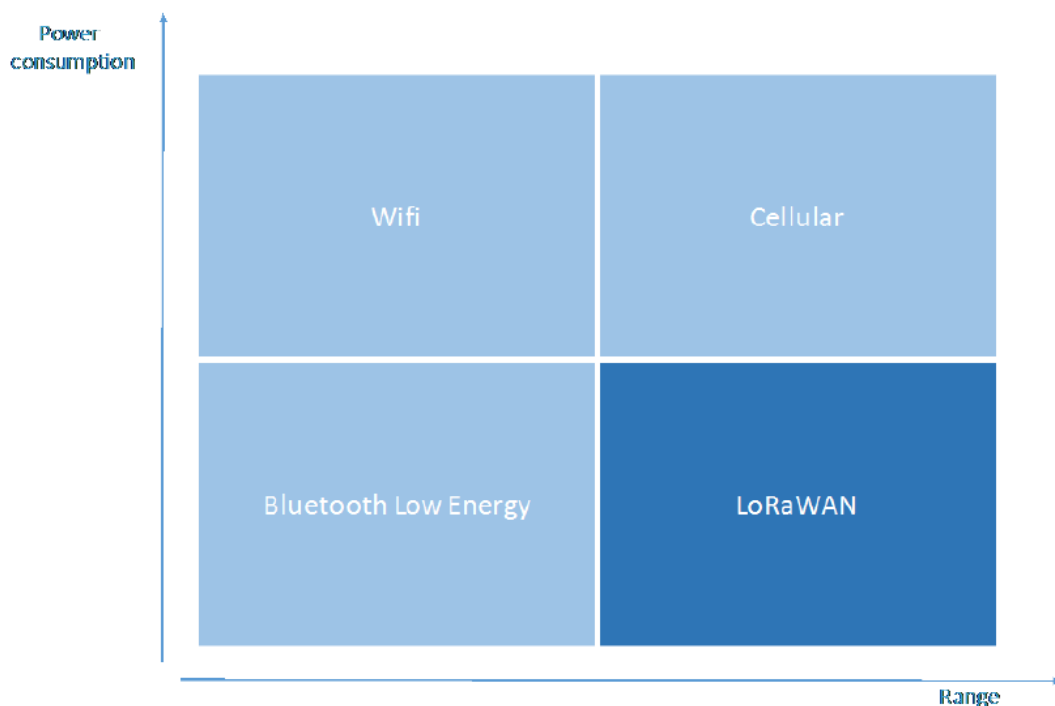
- Sigfox
- LoRa

Vhodnost použití zmíněných technologií ukazuje obr. 2.12.

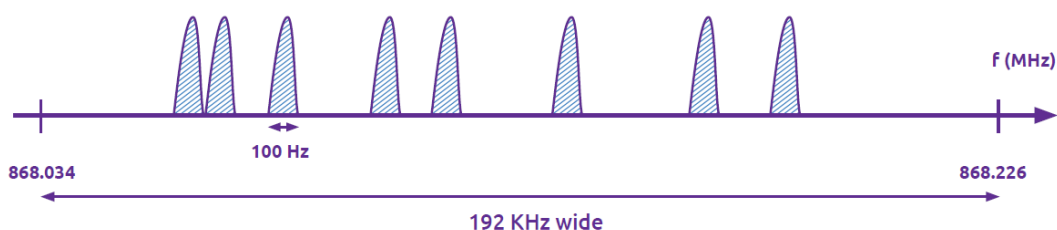
2.4.1 Sigfox

Sigfox je technologie určená pro zařízení jako jsou např. elektroměry, chytré hodinky, automatické pračky - tedy zařízení, které odesílají malé množství dat.

Sigfox využívá bezlicenční ISM pásma. V Evropě využívá frekvenční pásmo 868MHz, viz. obr.2.13 a v USA 915MHz. Denně je možno odeslat 144 zpráv o velikosti až 12byťů. Dosah se udává až 50km ve venkovských oblastech a cca 3km v



Obr. 2.12: IOT - Technologie v závislosti na dosahu a odběru energie[9]



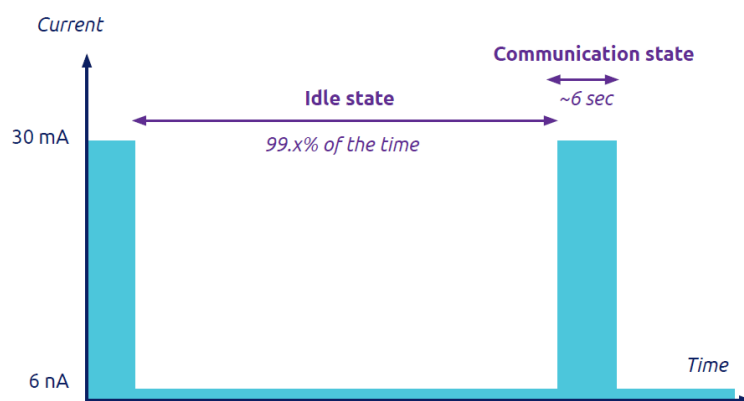
Obr. 2.13: Vysílací pásma sigfox[12]

oblastech městských. Odběr proudu při vysílání je 10mA až 50mA. Dlouhá životnost při napájení z baterie se dosahuje tím, že zařízení je 99% času ve stavu idle, kde je spotřeba minimální a co nejkratší čas ve stavu vysílání.[12]

Typický průběh odběru proudu je na obr.2.14. Pro zapojení do sítě je třeba se registrovat u lokálního operátora, např. v České republice SimpleCell.

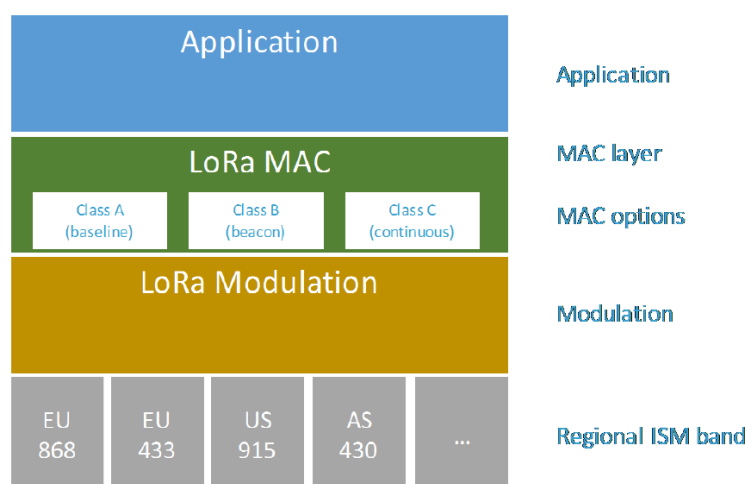
2.4.2 LoRa

České radiokomunikace umožňují zapojení do sítě technologie LoRa. LoRa podobně jako Sigfox využívá frekvenční pásmo 868MHz. Oproti Sigfox nabízí teoreticky neomezený počet zpráv denně, přičemž velikost zprávy může být až 255bytů. Dosah se udává 40km v oblastech venkovských a 2-5km v oblastech městských.



Obr. 2.14: Průběh odběru proudu - sigfox[12]

LoRa pracuje pouze na fyzické vrstvě. Při komunikaci se využívá komunikačního protokolu LoRaWan pracujícího na podvrstvě MAC(Media Access Control) linkové vrstvy ISO/OSI modelu, viz. obr.2.15.[9]



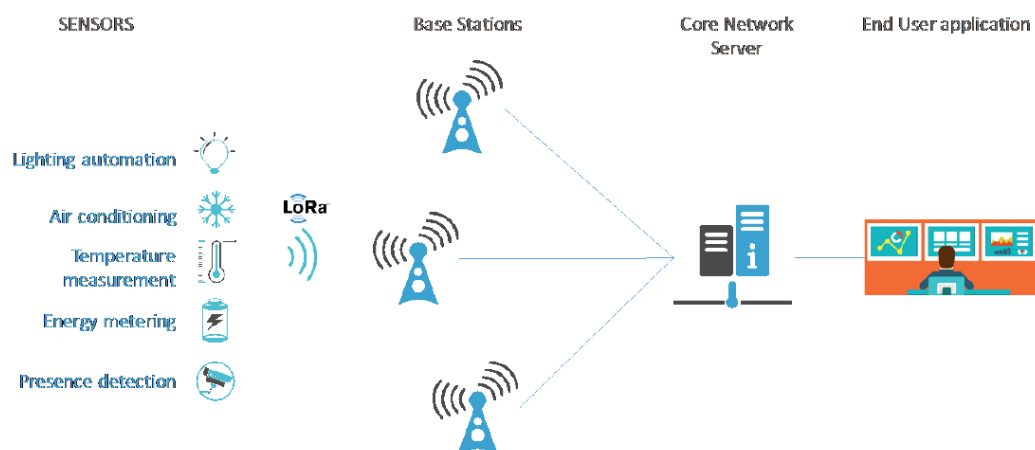
Obr. 2.15: ISO/OSI - LoRa[9]

Na obr.2.16 je zobrazena typická topologie LoRaWan sítě. BTS-Base Stations přenáší zprávy mezi koncovým zařízením a Core network serverem, přičemž všechny BTS jsou připojeny k serveru pomocí standardního IP připojení. Koncová zařízení využívají LoRa komunikaci s jednou nebo s více BTS. Přirozeně je komunikace obousměrná, ale počítá se především s variantou, kdy přenos z koncového zařízení do serveru je dominantní.[9]

V LoRaWan síti se zařízení dělí do několika tříd:

- Třída A

Tuto třídu musí podporovat všechna zařízení. Vysílání zahájí koncový prvek. Koncové zařízení po odeslání zprávy otevírá 2 přijímací okna ve kterých



Obr. 2.16: Topologie LoRaWan sítě[9]

může dostat data nebo příkaz od serveru. Zařízení třídy A mají nejnižší odběr energie.[9]

- **Třída B**
Tato třída podporuje třídu A, ale navíc přidává naplánovaná přijímací okna, kdy zařízení čeká na zprávu ze serveru. Vhodné použití je v zařízeních, které potřebují rychlejší odezvu nebo častější obdržení příkazu ze serveru.[9]
- **Třída C**
Třída C zahrnuje zařízení, které většinou mají externí zdroj napájení, tudíž požadavek na co nejnižší odběr energie není podstatný. Tato zařízení mají otevřeno přijímací okno vždy, pokud nevysílají zprávu.[9]

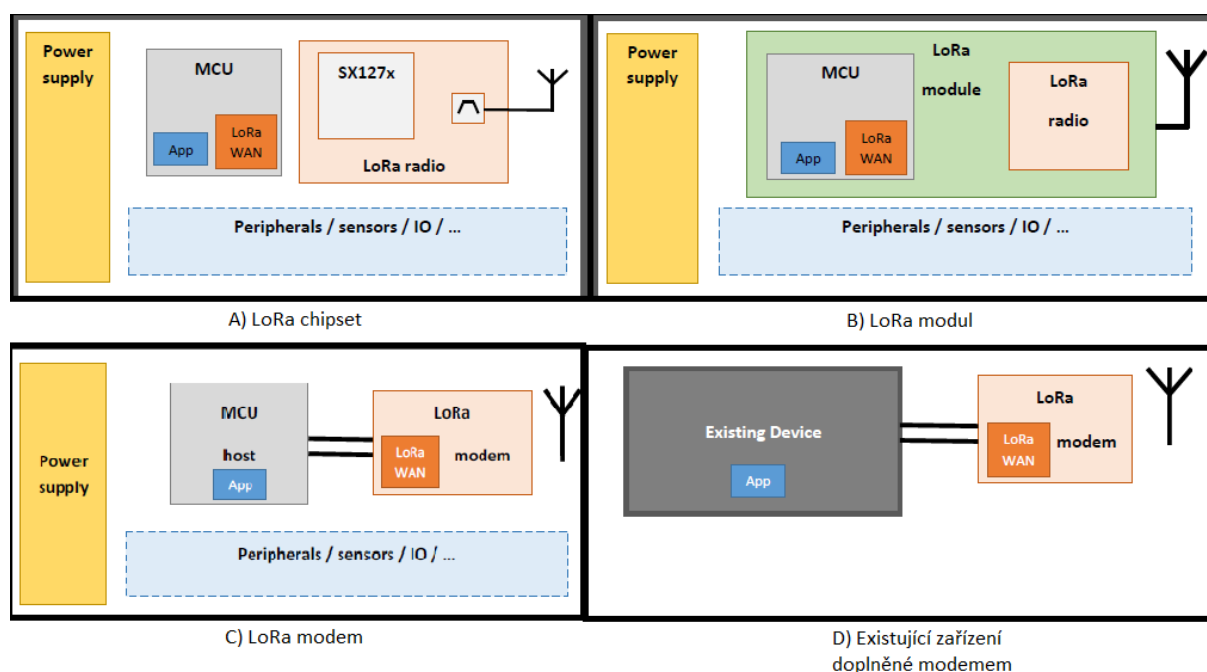
V této práci bude využita technologie LoRa. Jedním z důvodů tohoto výběru je teoretický neomezený počet zpráv denně. Druhým důvodem je operátor. České radiokomunikace mají oproti firmě SimpleCell větší možnost "přežití" na trhu vzhledem ke skutečnosti, že pokrývají vícero oblastí trhu, zatímco firma SimpleCell závisí na technologii Sigfox, což může být určující s příchodem nových technologií. Toto je ovšem pouze osobní názor.

Vzhledem ke skutečnosti, že zařízení vyvíjené v této práci je třeba optimalizovat pro co nejnižší odběr energie, bude zařízení využívat třídu A.

V případě vývoje LoRa zařízení existuje několik možností, zobrazeno na obr. 2.17:

- **Design založený na LoRa chipsetu**
Zařízení obsahuje LoRa vysílač, LoRaWan musí implementovat výrobce zařízení na svém mikrokontroléru a musí se též postarat o kompletní návrh RF hardwaru

- Design založený na LoRa modulu
LoRa modul obsahuje mikrokontrolér a LoRa vysílač. Není potřeba starat se o vývoj RF hardwaru, pouze je potřeba připojit anténu.
- Design založený na LoRa modemu
LoRa modem obsahuje LoRaWan i vysílací část. Některé modemy integrují i anténu. Modem se chová jako slave vůči nadřazenému mikrokontroléru se kterým komunikuje pomocí UART, USB nebo SPI.
- Vybavení existujícího zařízení LoRa modemem
Případ podobný případu předchozímu. Rozdíl je, že se např. při návrhu zařízení s IOT nepočítalo, proto je LoRa modem připojen externě - modem není na desce plošného spoje zařízení.



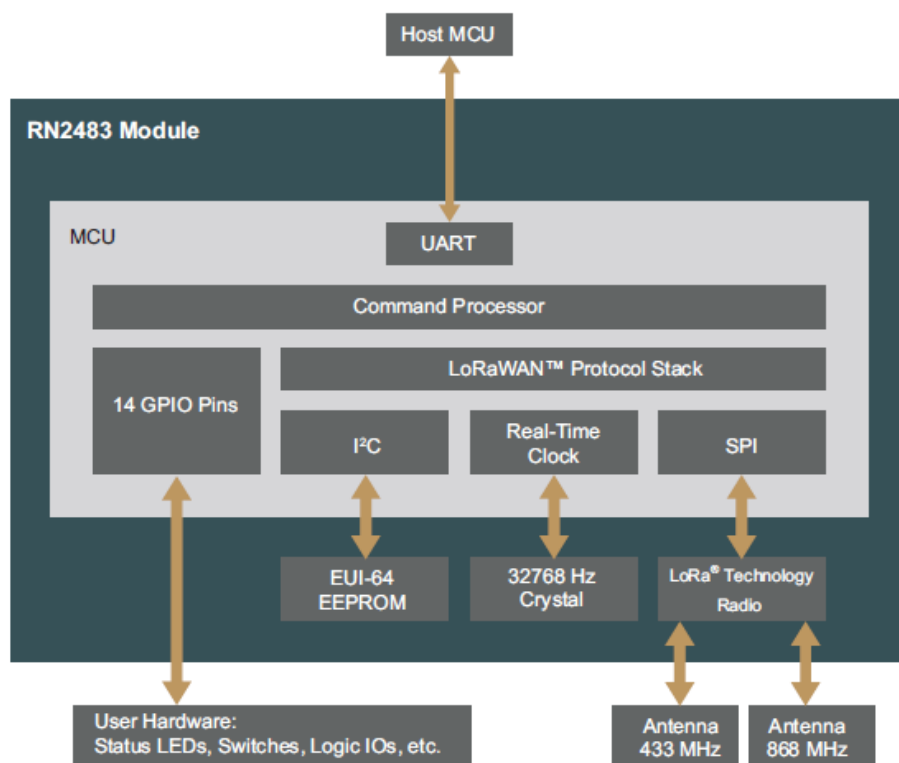
Obr. 2.17: Design LoRa zařízení[9]

V této práci bude použitý design založený na LoRa modemu. Konkrétně bude využit obvod RN2483 od firmy Microchip Technology. Blokové schéma je vidět na obr.2.18. Obvod RN2483 s nadřazeným mikrokontrolérem komunikuje pomocí UART.

2.5 Mikrokontrolér

Mikrokontrolér použitý v této práci bude od firmy STMicroelectronics, a to z několika důvodů:

- Poměr cena/výkon



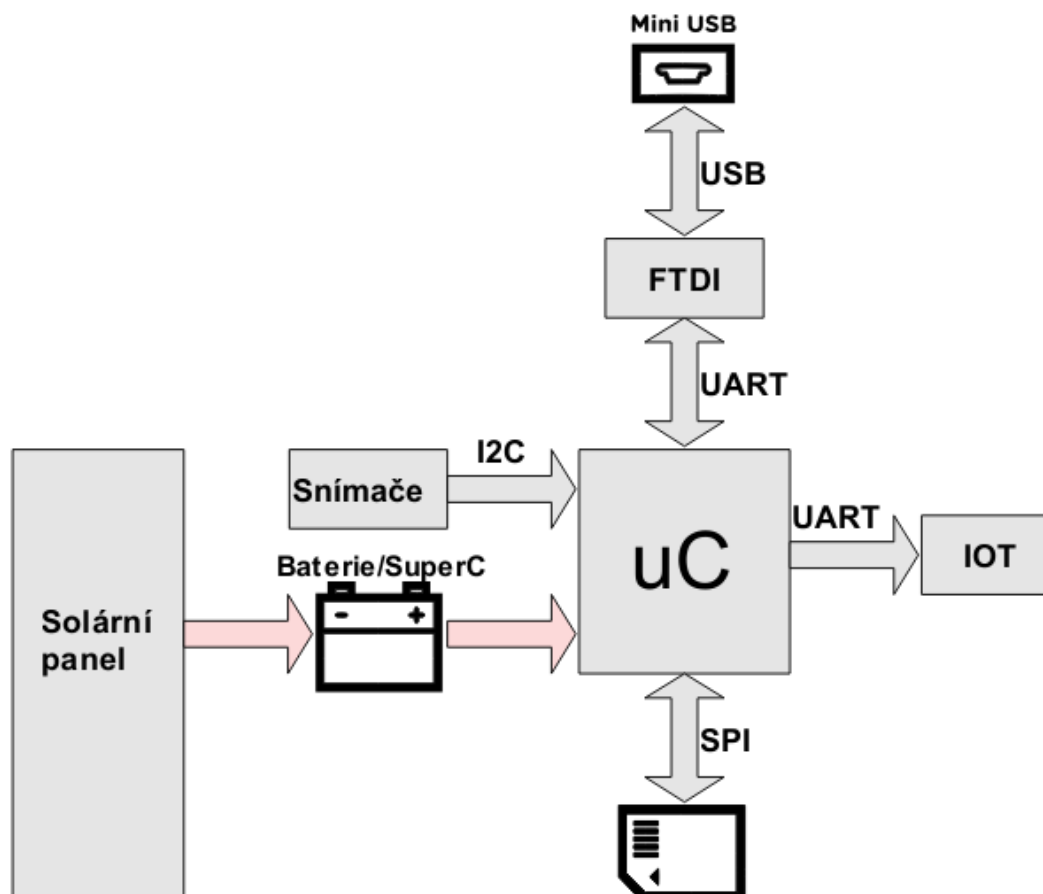
Obr. 2.18: Struktura RN2483[10]

- Dostupnost vývojových kitů a materiálů k učení
- Podpora nízkoenergetického designu
- 32bit. mikrokontroléry jsou v této době standard v průmyslu
- Autorova zkušenost s těmito mikrokontroléry

Po přečtení předchozích kapitol je možno upravit obrázek konceptu záznamníku o sběrnice, které budou potřeba, viz.obr. 2.19 a specifikovat požadavky na mikrokontrolér:

- Nízký odběr energie, podpora Low Power módů
- Přítomnost I2C pro připojení snímačů
- Přítomnost SPI pro připojení paměťové karty
- Přítomnost USART pro obvod FTDI a LoRa modul
- Přítomnost ADC pro měření napětí baterie/superkapacitoru
- Dostatečně velká kapacita paměti RAM pro dočasné ukládání naměřených dat
 - zápis na paměťovou kartu bude prováděn v určitých intervalech po dávkách

Pro aplikace u kterých je kladen důraz na nízký odběr energie nabízí firma STMicroelectronics 32bit. mikrokontroléry řady L. Po průzkumu trhu byl vybrán mikrokontrolér s označením STM32L151. Základní informace o tomto mikrokontro-



Obr. 2.19: Blokové schéma záznamníku doplněné o sběrnice

léru jsou uvedeny v tab. 2.2. Podrobnosti je možno dohledat v datasheetu mikrokontroléru.

Napájecí napětí [V]	Provozní teplota [°C]	Flash paměť [kB]	RAM paměť [kB]	Spotřeba Run-mode uA/MHz	Cena [Kč]
1.65 - 3.6	-40 - 105	384kB	80kB	195	230

Tab. 2.2: Základní parametry mikrokontroléru STM32L151

Mikrokontrolér poskytuje následující komunikační rozhraní:

- 5 USART
- 8 SPI
- 2 I2C
- 1 USB

2.6 Energetická bilance

Energetická bilance je zpracována na základě následujících parametrů:

- Odběr mikrokontroleru v režimu spánku je 20uA
- Průměrný odběr při měření dat je 1mA po dobu 1s, měření probíhá maximálně 1x za minutu
- Průměrný odběr SD karty při zápisu je 40mA po dobu 2s, zápis probíhá jednou za 10min.
- Průměrný odběr při vysílání dat je 40mA po dobu 3s, vysílání probíhá jednou za 10min.

Popsané parametry jsou zaneseny do tab. 2.3 a jsou využity pro výpočet průměrného odběru proudu.

Operace	Odběr [mA]	Doba trvání [ms]	Počet/10min	Odběr/10min [A]
Měření	1	1000	10	0.01
Vysílání	40	3000	1	0.12
Zápis na SD	40	2000	1	0.08
Režim spánku	0.02	585000	1	0.0117

Tab. 2.3: Energetická spotřeba jednotlivých operací

Celkový odebíraný proud za 10min je součtem proudových odběrů za 10min, uvedených v posledním sloupci tab. 2.3:

$$I_c = I_{mer} + I_{Vys} + I_{zap} + I_{spanek} = 0.01 + 0.12 + 0.08 + 0.0117 = 0.2217A/10min \quad (2.1)$$

Průměrně odebíraný proud získáme vydělením celkově odebíraného proudu časem:

$$I_{avg} = \frac{I_c}{t} = \frac{0.2217}{600} = 0.37mA \quad (2.2)$$

Ze znalosti průměrně odebíraného proudu a napájecího napětí 3.3 V můžeme určit průměrný výkon:

$$P_{avg} = I_{avg} \cdot U = 0.37mA \cdot 3.3V = 1.23mW \quad (2.3)$$

Abychom získali zmiňovaných 3.3V, tak bude potřeba step-up měnič - v případě napájení ze superkapacitoru i v případě napájení z baterie. Účinnost měniče budeme uvažovat 80%. Tím pádem se potřebný průměrný výkon zvýší následujícím způsobem:

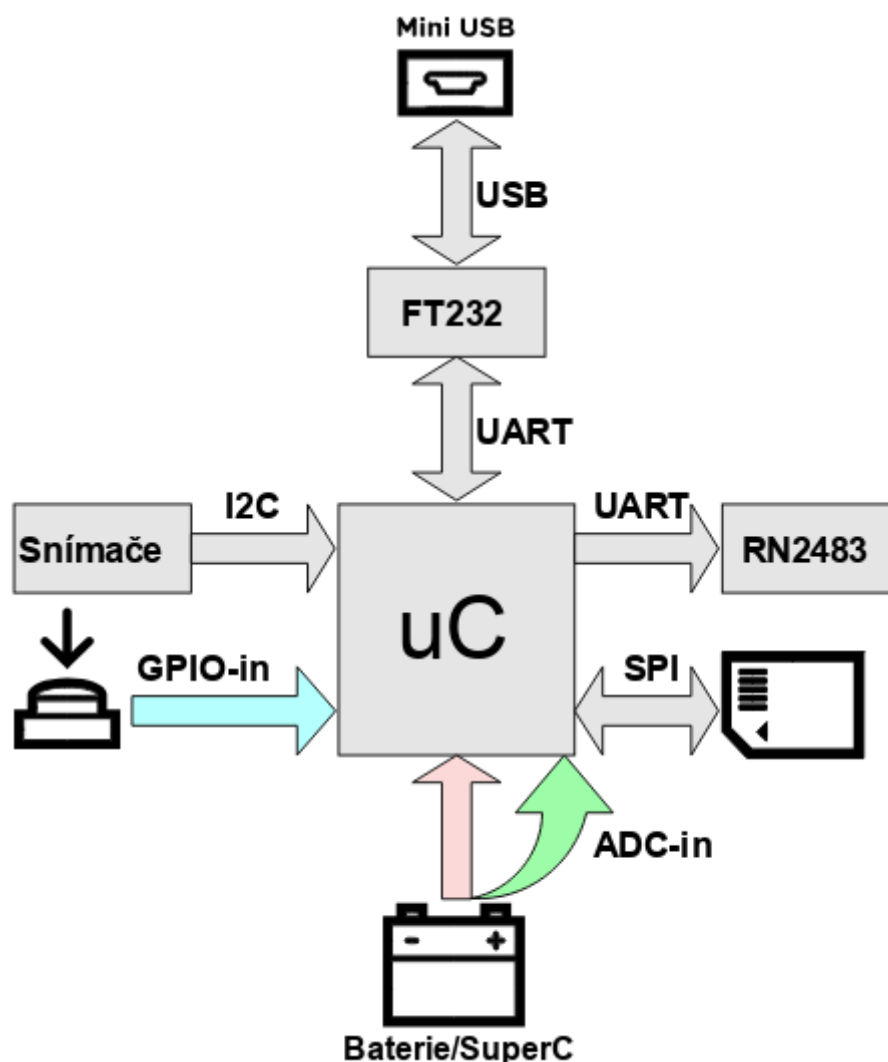
$$P_{avg} = P_{avg} \cdot \frac{100}{80} = 1.53mW \quad (2.4)$$

3 Návrh HW

V této části bude popsán návrh HW - tedy schématu a desky plošných spojů hlavní mikrokontrolérové části a napájecí části.

3.1 Návrh hlavní desky

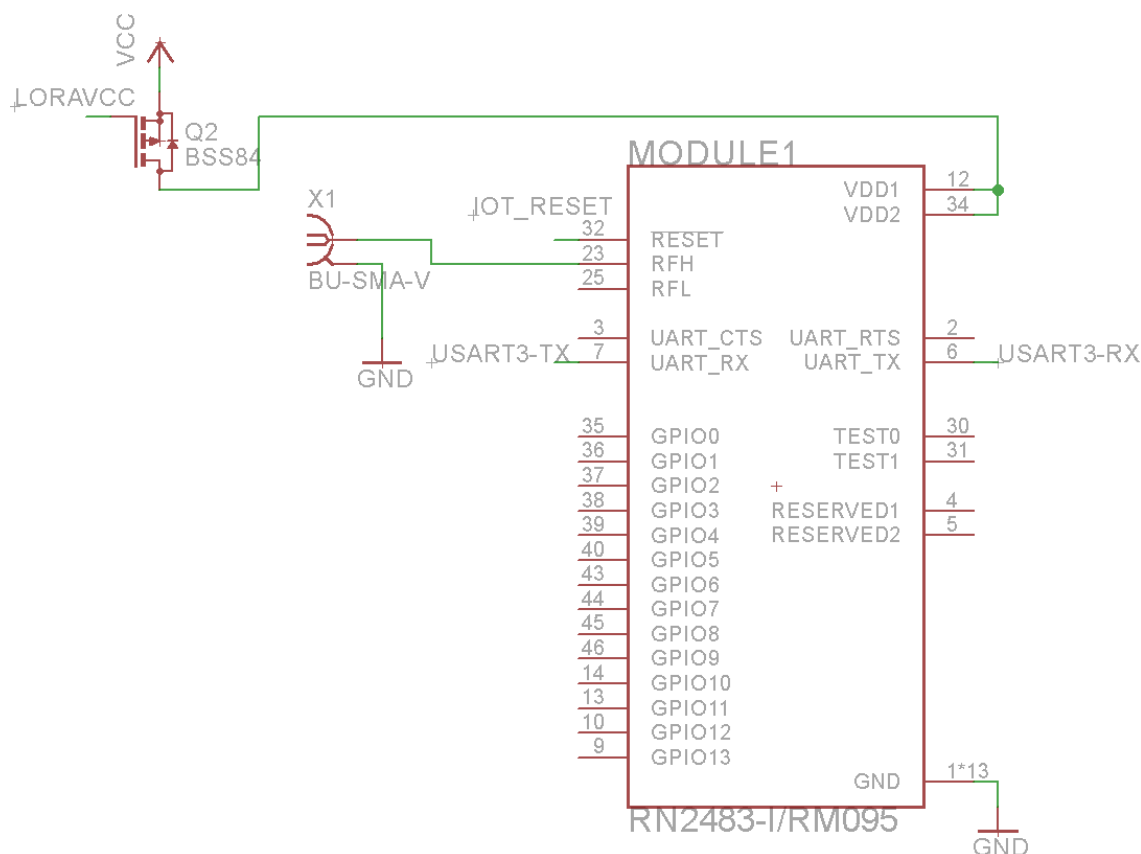
Označením hlavní desky je myšlena část zařízení, která obsahuje mikrokontrolér, obvod pro komunikaci přes USB, paměťovou kartu, obvod pro vysílání dat pomocí LoRa a snímače pro měření klimatických dat. Dále je potřeba měřit napětí na superkapacitoru/baterii. Pro konfiguraci zařízení budou k dispozici 3 tlačítka. Blokové schéma hlavní desky je zobrazeno na obr. 3.1.



Obr. 3.1: Blokové schéma hlavní desky

3.1.1 Schéma zapojení IOT modulu RN2483

Schéma zapojení modulu RN2483 je velice jednoduché. V podstatě je potřeba jen připojit UART piny Tx, Rx pro komunikaci s mikrokontrolérem, anténu a pin reset na výstup mikrokontroléru, aby bylo možné modul v případě potřeby resetovat. Schéma je zobrazeno v obr. 3.2.



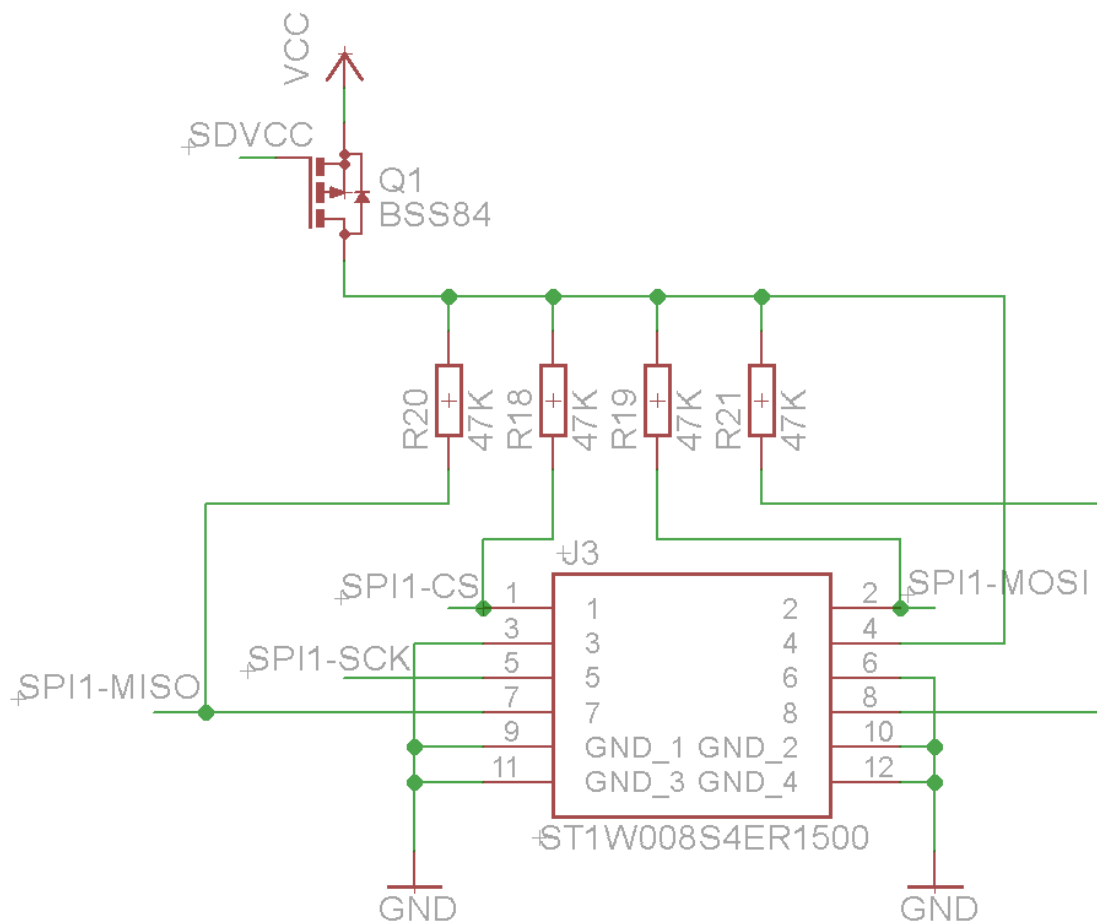
Obr. 3.2: Schéma zapojení IOT modulu RN2483

3.1.2 Schéma zapojení SD karty

Pro komunikaci s SD kartou slouží SPI sběrnice. Kvůli úspoře místa na dps byl využit konektor pro microSD karty. Pro udržení vysoké logické úrovně v klidovém stavu jsou připojeny na příslušné piny pull-up rezistory. Schéma je zobrazeno v obr. 3.3.

3.1.3 USB komunikace

Na obr. 3.4 je zobrazeno schéma zapojení obvodu FT232RL. Jedná se o katalogové zapojení obvodu s připojením k mikrokontroléru a napájením z USB sběrnice. Oproti

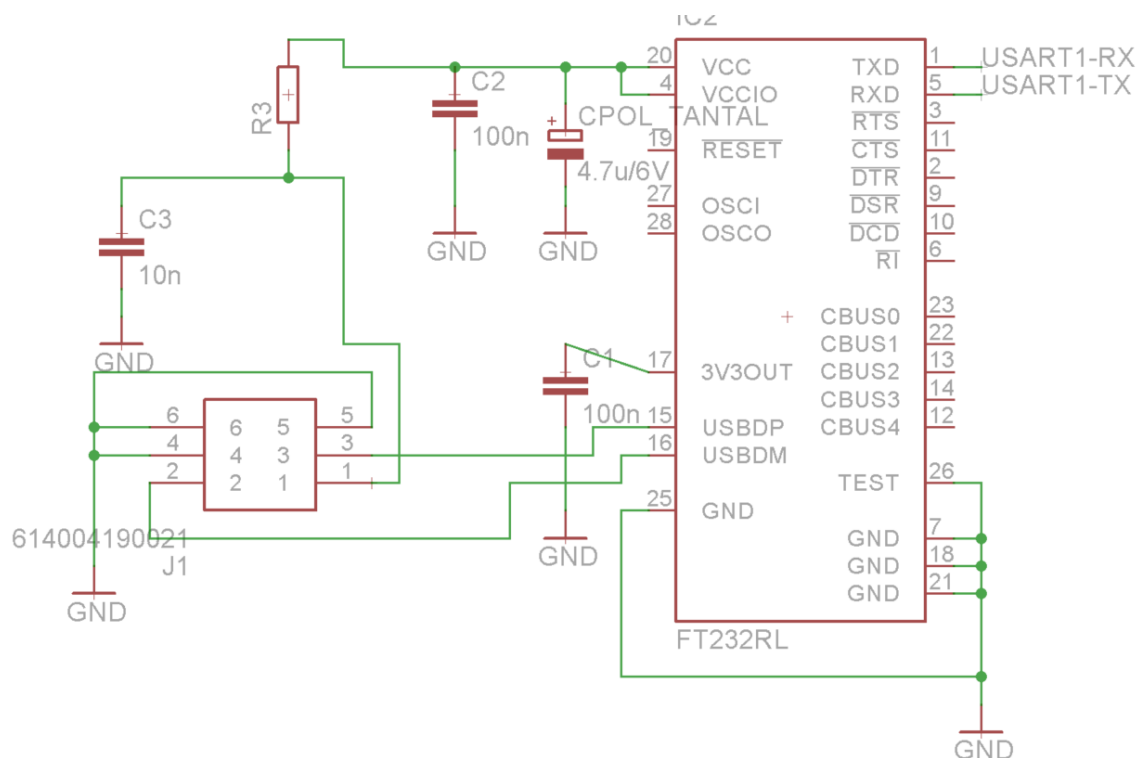


Obr. 3.3: Schéma zapojení SD karty

katalogovému zapojení je třeba si dát pozor a připojit pin TEST na GND, jinak se zařízení bude v PC hlásit jako poškozené. R3 ze schématu představuje feritovou perlu sloužící jako prevence proti EMI - Electromagnetic interference. K fyzickému propojení PC a zařízení byl využit USB konektor typu A.

3.1.4 Mikrokontrolérová část

K napájecím pinům byly připojeny 100nF kondenzátory dle doporučení z ref. manualu [14]. Dále byl k mikrokontroléru připojen 32kHz krystal sloužící jako zdroj impulsů pro RTC obvod mikrokontroléru. Ke krystalu jsou připojeny kondenzátory dle doporučení výrobce krystalu. Pin BOOT0 byl připojen na GND, protože budeme mikrokontrolér programovat přes rozhraní SWD a kód bude obsažen ve Flash paměti. Dále byly alokovány piny pro sběrnice I2C, SPI a 2x UART. Piny které mohou náležet těmto sběrnicím byly vybrány pomocí programu STM32CubeMX, přičemž to bylo pro jistotu zkontrolováno s tabulkou pro alternate function z ref manualu mikrokontroléru[14]. Pro udržení vysoké logické úrovně v klidovém stavu



Obr. 3.4: Schéma zapojení části určené pro USB komunikaci

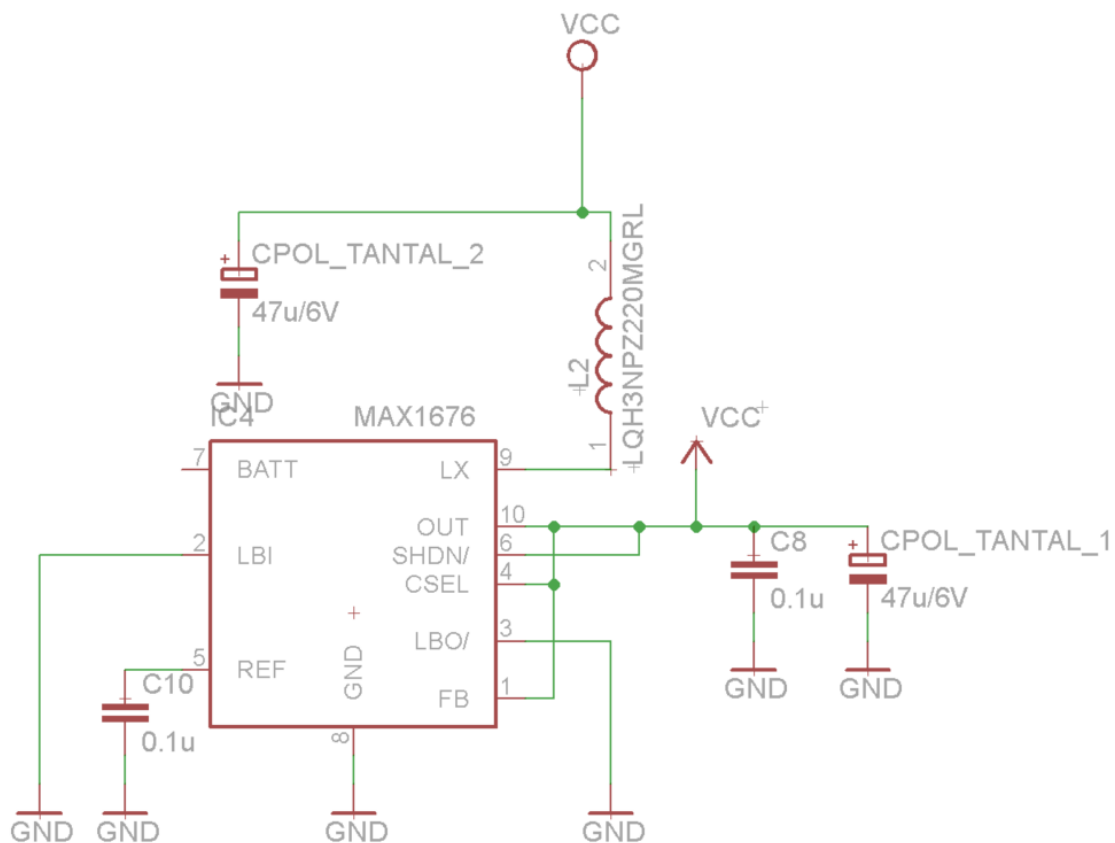
jsou k pinům pro I2C sběrnici připojeny pull-up rezistory. Dále byla deska vybavena konektorem určeným pro připojení programátoru přes rozhraní SWD. Nakonec bylo vyhrazeno několik GPIO pinů sloužících pro následující účely:

- Vstup AD převodníku
- Připojení tlačítka
- Ovládání připojení nebo odpojení napájecího napětí od IOT modulu, snímačů, SD karty

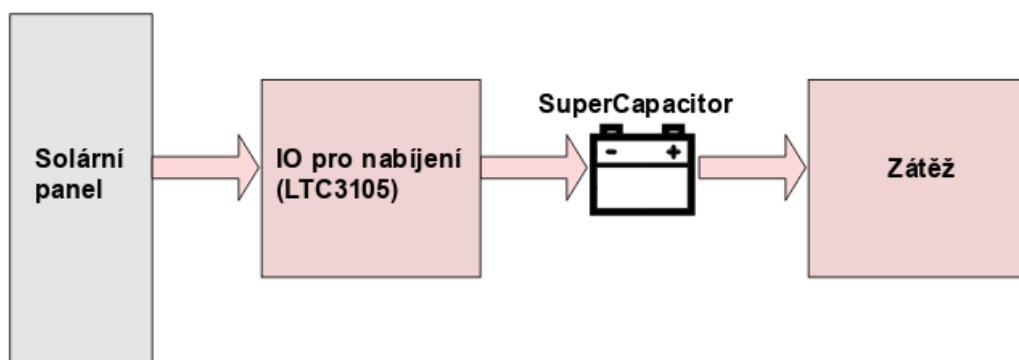
Schéma mikrokontrolérové části je na obr. 3.5.

3.1.5 Schéma napájecí části

Prvky hlavní desky potřebují napájecí napětí o velikosti 3,3V. Protože solární panel bude dobíjet superkapacitor nebo baterii, které budou mít napětí nižší než 3,3V je potřeba využít zvyšující měnič na 3,3V. K tomuto účelu byl využit integrovaný obvod MAX1676. MAX1676 je vysoce efektivní zvyšující měnič, který je schopen pracovat už od vstupního napětí 1,1V. Výstupní napětí je možno připojením pinu FB na pin OUT nebo GND nastavit na 3,3 nebo 5V. Bylo tedy použito katalogové zapojení s připojením pinu FB na pin OUT. Zapojení je zobrazeno na obr. 3.6.



Obr. 3.6: Schéma napájecí části



Obr. 3.7: Blokové schéma zdroje

budeme superkapacitor provozovat v rozsahu napětí 1.3V - 3V. Uvažujeme tedy rezervu 0.1V.

Proud kondenzátorem je dán následující rovnicí:

$$i_c = C \frac{dV_c}{dt} \quad (3.1)$$

Proud se ze znalosti výkonu a napětí určí dle následující rovnice:

$$i = \frac{P}{U} \quad (3.2)$$

Předchozí dvě rovnice se mohou upravit do následujícího tvaru:

$$\frac{P}{U} = C \frac{dU_c}{dt} \quad (3.3)$$

Následně osamostatníme dU a přidáme integrál:

$$\int dU_c = \frac{1}{U_c C} \int P dt \quad (3.4)$$

Poté rovnici upravíme:

$$\int_{U_{max}}^{U_{min}} U_c dU_c = \frac{P}{C} \int_T^0 dt \quad (3.5)$$

Následně rovnice pro výpočet času po který je kapacitor schopen dodávat energii:

$$t = \frac{0.5C(U_{max}^2 - U_{min}^2)}{P} \quad (3.6)$$

Rovnice upravená pro určení hodnoty superkapacitoru a dosazení:

$$C = \frac{tP}{0.5(U_{max}^2 - U_{min}^2)} = \frac{86400 \cdot 0.00153}{0.5(2.9^2 - 1.3^2)} = 39.4F \quad (3.7)$$

Odvození a výpočet potřebné kapacity byl ověřen pomocí simulačního softwaru LTspice. Na obr. 3.8 je simulační schéma, kde je kondenzátor spolu s konstantní zátěží a na obr. 3.9 je výsledek simulace. Počáteční napětí na kondenzátoru je nastaveno na 2.9 V a po 24h klesne na 1.3 V, což potvrzuje předchozí odvození a výpočet potřebné kapacity superkapacitoru.

Samozřejmě superkondenzátor s hodnotou kapacity 39.4 F koupit nelze. A proto bude použit superkondenzátor s kapacitou 50 F, čímž zároveň bude zaručena 25% rezerva.

Pokud uvažujeme, že v zimních bude dostatek světla pro dobíjení superkapacitoru v denní době od 10h do 15h, bude potřeba dobíjet superkapacitor proudem:

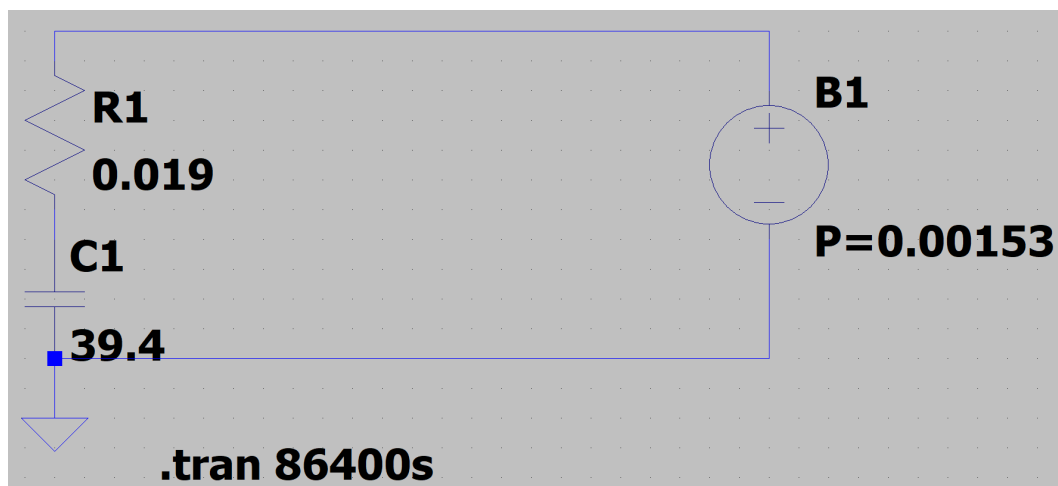
$$I_{dob} = \frac{C(U_{cmax} - U_{cmin})}{t} = \frac{50(2.9 - 1.3)}{18000} = 4.5mA \quad (3.8)$$

3.2.2 Měření solárního článku

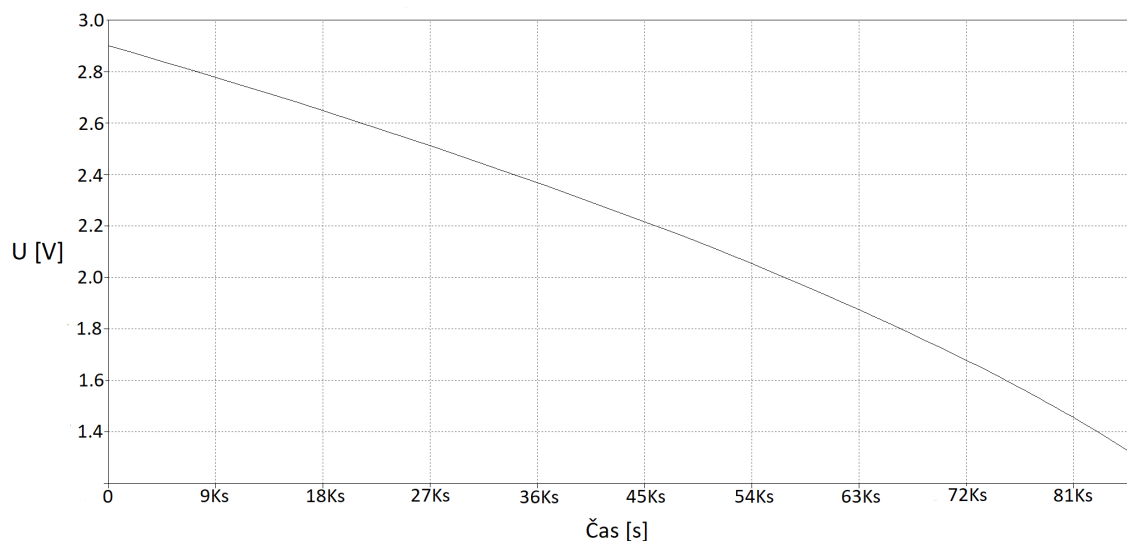
Bylo provedeno měření VA charakteristiky solárního článku, mmj. za účelem ověření, že solární článek bude schopen dobíjet superkapacitor.

Maximální velikost solárního článku byla zvolena, s ohledem na předpokládanou velikost zařízení a s ohledem na celkovou kompaktnost celku, jako velikost odpovídající velikosti papíru formátu A4, tj. 297×294 mm.

Parametry měřeného solárního článku jsou uvedeny v následujícím výčtu:



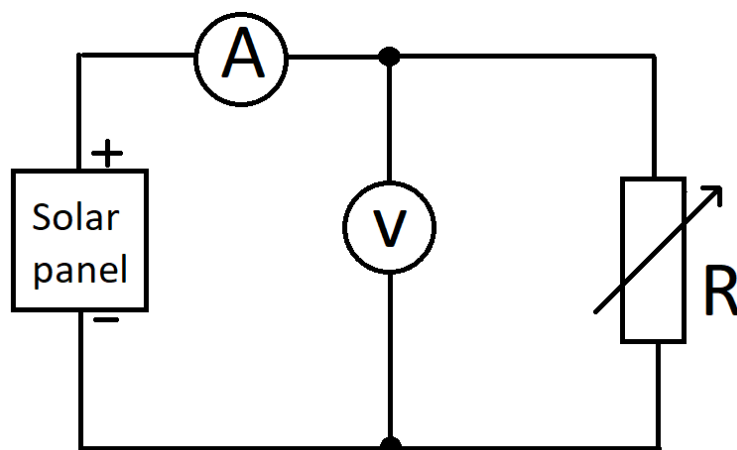
Obr. 3.8: Simulační schéma



Obr. 3.9: Průběh napětí na kondenzátoru

- Typ panelu - Amorfni
- Napětí naprázdno - 7V
- Výkon - 2W
- Zkratový proud - 286mA
- Rozměr - 320 × 160mm

Měření proběhlo 2×, dle schématu 3.10. Obojí měření proběhlo v měsíci listopad 2019 v čase po 10. hodině ranní. Jednou měření proběhlo za relativně dobrých světelných podmínek s ohledem na podzimní/zimní měsíc, tj. jasné počasí. Jednou měření proběhlo za velmi špatných podmínek, konkrétně zataženo s lehkým deštěm. Naměřená VA charakteristika je vykreslena v obr. 3.11. Nahoře je měření odpovídající jasnému počasí, dole je měření odpovídající zataženému počasí s deštěm.



Obr. 3.10: Schéma měření VA charakteristiky solárního článku

Z grafu 3.11 dole je jasné, že solární článek není schopen dobít superkapacitor za špatných světelných podmínek. Je tedy potřeba změnit koncept zdrojové části, pravděpodobně využít baterii jako hlavní zdroj energie pro zařízení nebo využít baterii jako záložní zdroj energie.

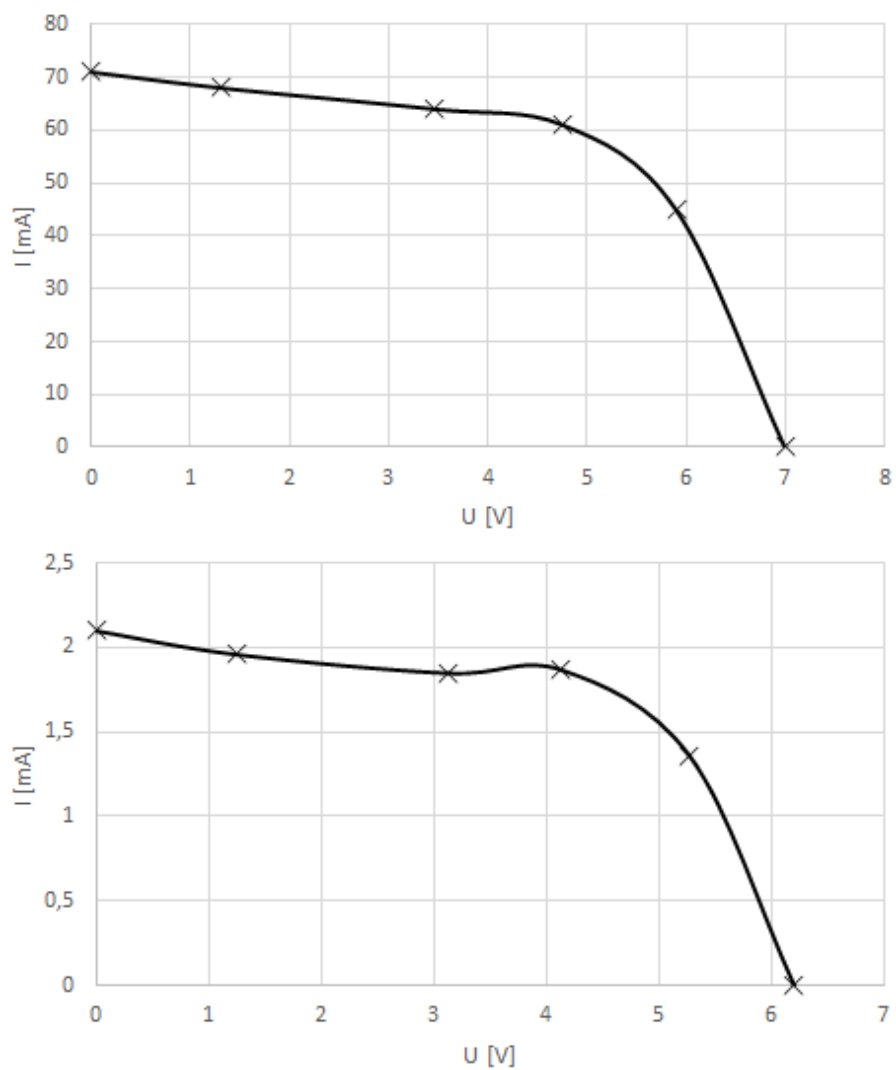
3.2.3 Koncept 2 - baterie i superkapacitor

Solární článek o maximální akceptovatelné velikosti, tj. 297x210 mm, není schopen za neideálních světelných podmínek, tzn. zataženo, déšť nebo průměrně osvětlený vnitřní prostor, dobít superkapacitor. A proto byl navržen koncept využívající Lithium-titanovou baterii jako v podstatě záložní zdroj, ze kterého bude zařízení provozováno ve výše zmíněných případech. Koncept zdroje je uveden na obr.3.12.

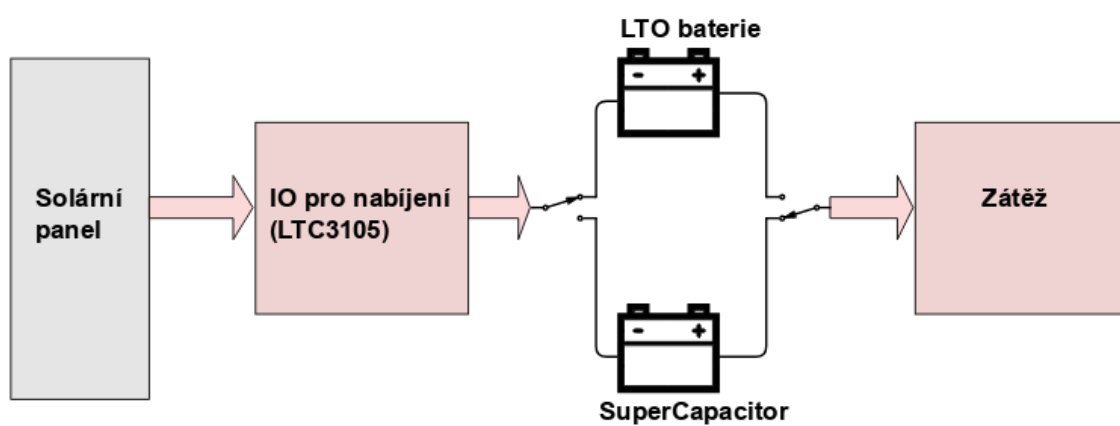
Základem budou v roli přepínačů 2 bistabilní dvojcívkové relé. Solární panel přes integrovaný obvod nabíjí superkapacitor nebo baterii. Hlavní deska zařízení je napájena ze superkapacitoru nebo z baterie. Přepínání relé bude ovládáno pomocí mikrokontroléru z hlavní desky přes tranzistor.

Jako dobíjecí obvod budou vytvořeny 2 varianty, pro použití se solárními články o různém napětí. Jedna varianta pro solární články s napětím do 5V a druhá varianta pro vyšší napětí.

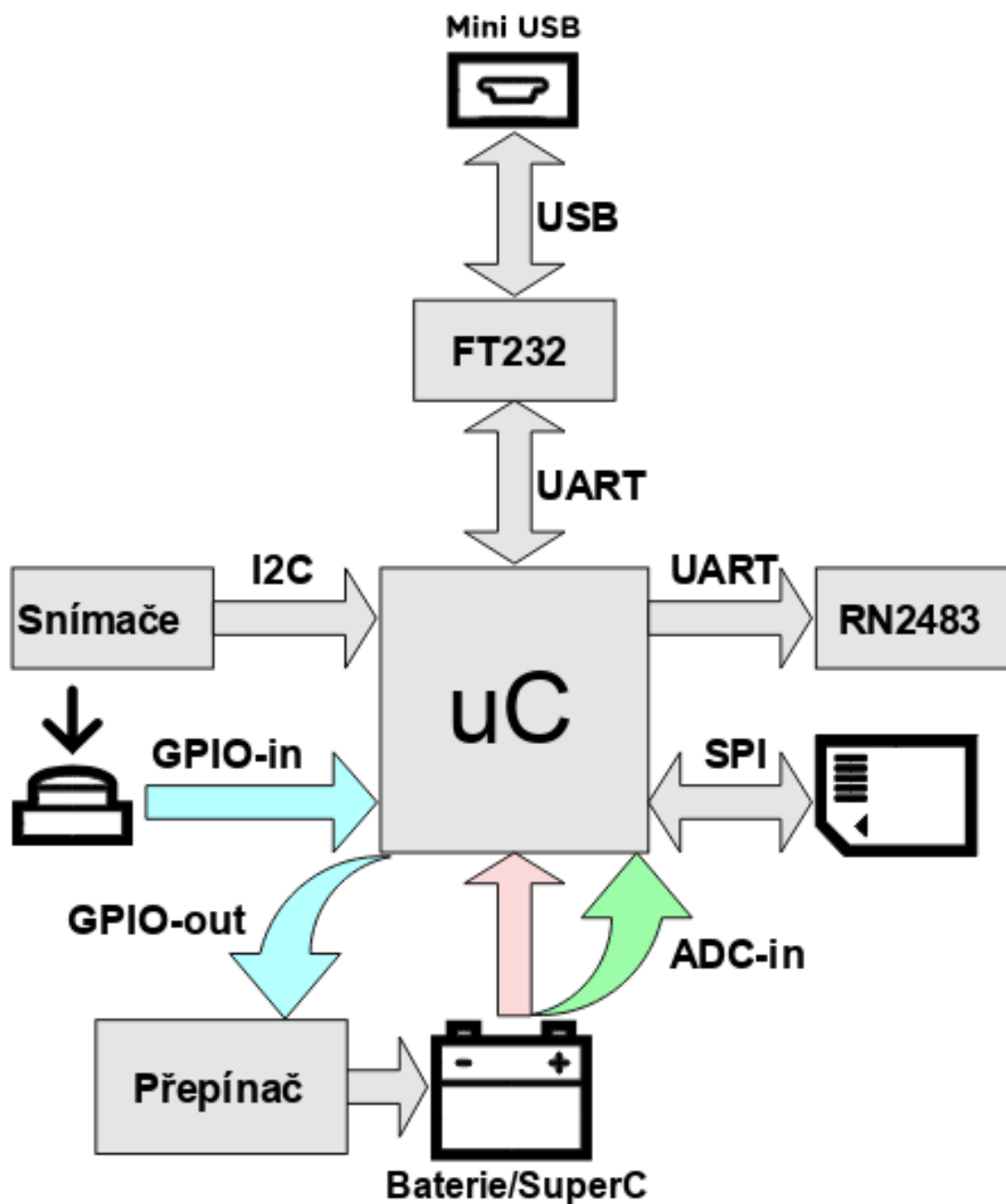
Se změnou konceptu zdroje je potřeba mírně modifikovat koncept hlavní desky. Konkrétně je třeba poskytnout GPIO pro ovládání přepínání mezi dobíjením superkapacitoru a baterie, a také přepínání mezi baterií a superkapacitorem ve funkci napájení hlavní desky. Blokové schéma je na obr. 3.13.



Obr. 3.11: VA char. solárního článku - jasno(nahoře), zataženo(dole)



Obr. 3.12: Blokové schéma nového konceptu zdroje



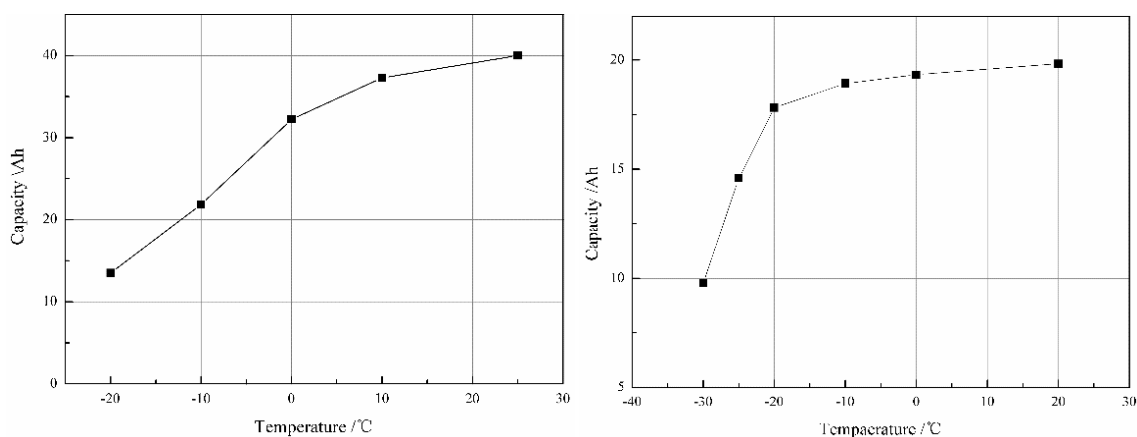
Obr. 3.13: Vylepšené blokové schéma hlavní desky

3.2.4 Koncept 3 - Lithium-Titanová baterie

Koncept 3 je podobný konceptu 1, pouze je superkapacitor nahrazen baterií. Tzn. že schéma, ovládání je stejné jako u konceptu 1.

Baterie má oproti superkapacitoru výhodu, že je schopna zařízení napájet i pokud několik dní, desítek dní nebude přístupno sluneční záření, které by muselo dobíjet superkapacitor. Nicméně baterie má horší vlastnosti, zejména počet nabíjecích/vy-

bíjecích cyklů a horší vlastnosti při provozu ve velmi nízkých teplotách. Na obr. 3.14 je zobrazena závislost dostupné kapacity dvou typů baterií na teplotě, konkrétně vlevo je závislost pro Lithiovou baterii a vpravo pro Lithium-titanovou baterii. Oba typy baterií by se daly koupit a použít v navrhovaném zařízení. Z obr. 3.14 plyne, že pro naše účely by mohla být použita Lithium-titanová baterie, která má oproti Lithiové baterii při teplotě -20°C poloviční pokles kapacity. Kromě nižšího poklesu kapacity má i vyšší počet nabíjecích/vybíjecích cyklů[13].



Obr. 3.14: Závislost kapacity baterie na teplotě - vlevo Lithium, vpravo Lithium-titan [13]

3.2.5 Srovnání konceptů zdroje

V této části práce je provedeno srovnání konceptů zdroje na základě výhod, nevýhod a je rozhodnuto o konkrétním výběru konceptu.

Koncept 1

V následujícím výčtu jsou sepsány výhody použití konceptu 1:

- Snadná implementace
- Počet cyklů nabíjení/vybíjení
- Schopnost pracovat i při velmi nízkých teplotách

V následujícím výčtu jsou sepsány nevýhody použití konceptu 1:

- Závislost na každodenním dobíjení ze slunce

Koncept 2

V následujícím výčtu jsou sepsány výhody použití konceptu 2:

- Využívá výhod napájení ze superkapacitoru/baterie dle aktuální situace

- Schopnost napájet zařízení po omezený čas i bez dostatečného slunečního záření - kdy není sluneční záření přístupno v řádu dnů, desítek dnů

V následujícím výčtu jsou sepsány nevýhody použití konceptu 2:

- Složité na implementaci
- Ekonomicky nejnáročnější koncept
- Zabere nejvíce prostoru oproti jiným konceptům

Koncept 3

V následujícím výčtu jsou sepsány výhody použití konceptu 3:

- Snadná implementace
- Schopnost napájet zařízení po omezený čas i bez dostatečného slunečního záření - kdy není sluneční záření přístupno v řádu dnů, desítek dnů

V následujícím výčtu jsou sepsány nevýhody použití konceptu 3:

- Horší vlastnosti baterie při nízkých teplotách oproti superkapacitoru

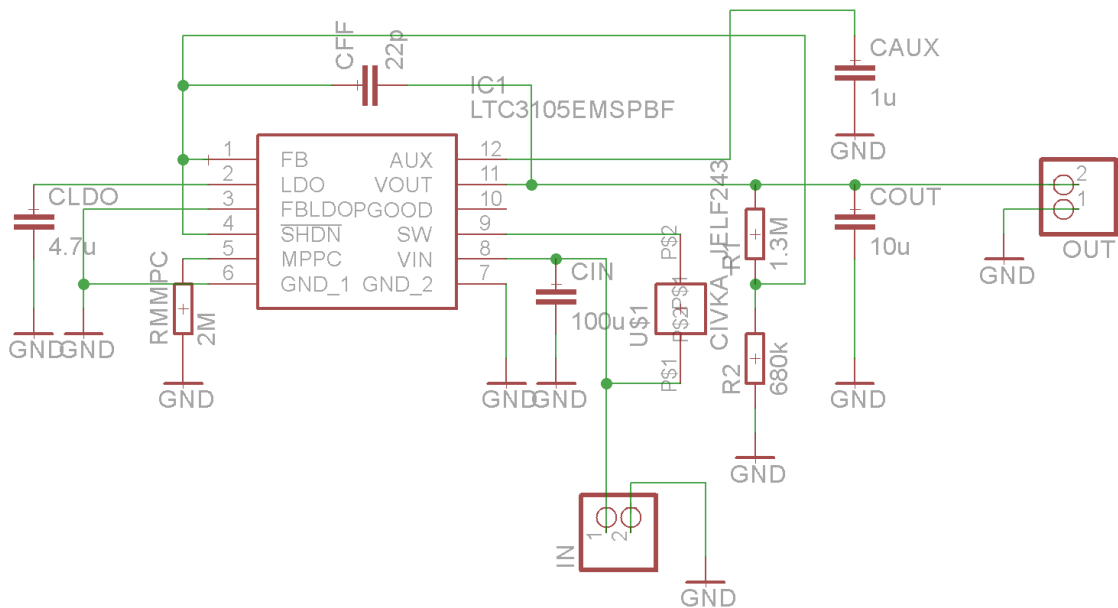
Bylo rozhodnuto, že bude použit koncept 3, tedy využití Lithium-titanové baterie, která je schopna uspokojivě pracovat i při nízkých teplotách, a zároveň jde i o výhodné řešení z hlediska ekonomického a z hlediska snadnosti implementace, při dosažení určité míry nezávislosti na slunečním záření.

3.2.6 Obvody pro dobíjení baterie

Byly vytvořeny 2 verze dps k dobíjení baterie. První verze s obvodem LTC-3105, který je schopen pracovat při vstupním napětí od 0.25V do 5V. Druhá verze s obvodem LTC-3129, který je určen pro vstupní napětí od 2.42V do 15V. Oba obvody jsou typu Buck-Boost, tzn. že jsou schopny pracovat s napětím nižším i vyšším než má být napětí výstupní.

LTC-3105

Schéma zapojení s obvodem LTC-3105 je na obr. 3.15. Jedná se o katalogové zapojení, s tím, že bylo potřeba určit hodnotu rezistorů R1, R2 na základě požadavku na výstupní napětí. Dále bylo potřeba určit hodnotu rezistoru RMMPC, na základě požadavku na hodnotu nejnižšího napětí solárního panelu, pod kterou napětí solárního panelu nemá klesnout - tzn. že se obvod nebude snažit zvýšit výstupní proud ze solárního panelu, čímž by došlo k poklesu jeho napětí - jedná se o tzv. funkci sledování maximálního výkonu.



Obr. 3.15: Schéma zapojení s obvodem LTC3105

Nabitá baterie má napětí cca 2.8V, a proto požadujeme výstupní napětí cca 2.8V. Výstupní napětí obvodu LTC3105 je dáno dle rovnice:

$$V_{out} = 1.004 \cdot \left(\frac{R1}{R2} + 1 \right) \quad (3.9)$$

R2 je doporučeno volit v řádu stovek $k\Omega$, volíme tedy $680k\Omega$. Následně pro R1 platí:

$$R1 = R2 \cdot \left(\frac{V_{out}}{1.004} - 1 \right) = 680k \cdot \left(\frac{2.8}{1.004} - 1 \right) = 1.26M\Omega \quad (3.10)$$

Chceme, aby napětí solárního panelu nekleslo pod úroveň 0.8V. VMPPC je dáno rovnicí:

$$V_{MPPC} = 10\mu A \cdot R_{MPPC} \quad (3.11)$$

Pro RMMPC pak platí:

$$R_{MPPC} = \frac{V_{MPPC}}{10\mu A} = \frac{0.8}{10\mu A} = 80k\Omega \quad (3.12)$$

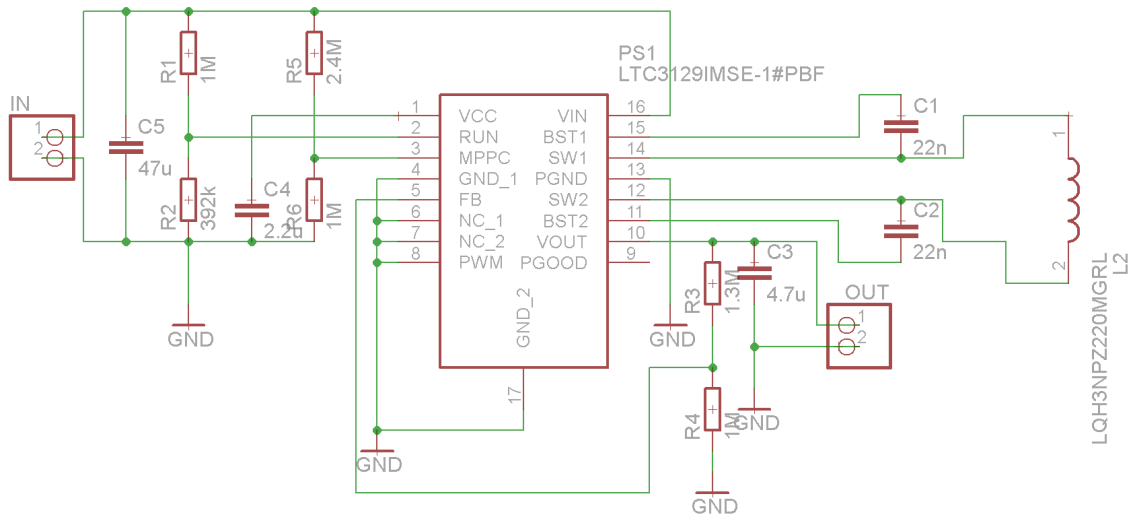
LTC - 3129

Schéma zapojení s obvodem LTC-3129 je na obr. 3.16. Nabitá baterie má napětí cca 2.8V, a proto požadujeme výstupní napětí cca 2.8V. Pro výstupní napětí obvodu LTC3129 platí rovnice:

$$V_{out} = 1.175 \cdot \left(\frac{R3}{R4} + 1 \right) \quad (3.13)$$

Hodnotu odporu rezistoru R4 volíme $1M\Omega$. Následně pro R3 platí rovnice:

$$R3 = R4 \cdot \left(\frac{V_{out}}{1.175} - 1 \right) = 1M \cdot \left(\frac{2.8}{1.175} - 1 \right) = 1.38M\Omega \quad (3.14)$$



Obr. 3.16: Schéma zapojení s obvodem LTC3129

Chceme, aby napětí solárního panelu nekleslo pod úroveň 4V. Pro VMPPC platí rovnice:

$$V_{MPPC} = 1.175 \cdot \left(\frac{R_5}{R_6} + 1 \right) \quad (3.15)$$

Hodnotu odporu R_6 volíme $1M\Omega$. Následně pro R_5 platí rovnice:

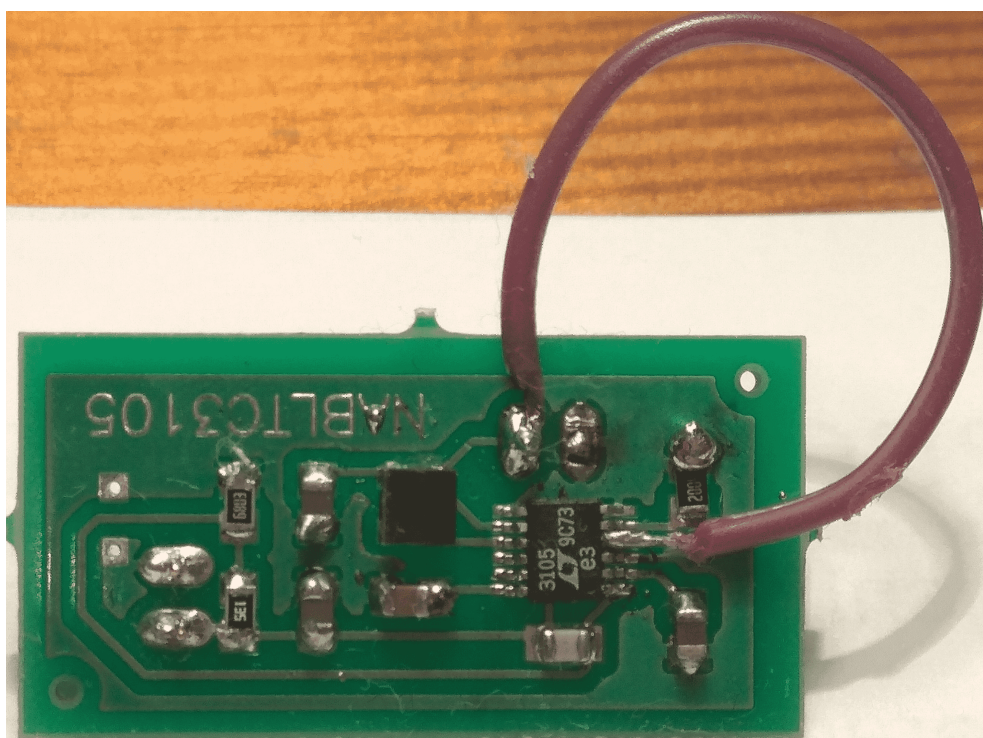
$$R_5 = R_6 \cdot \left(\frac{V_{MPPC}}{1.175} - 1 \right) = 1M \cdot \left(\frac{4}{1.175} - 1 \right) = 2.4M\Omega \quad (3.16)$$

4 Sestavení a oživení zařízení

V této části práce budou popsány kroky a nezbytné úpravy v zapojení, které postupně vedly k oživení zařízení.

4.1 LTC3105

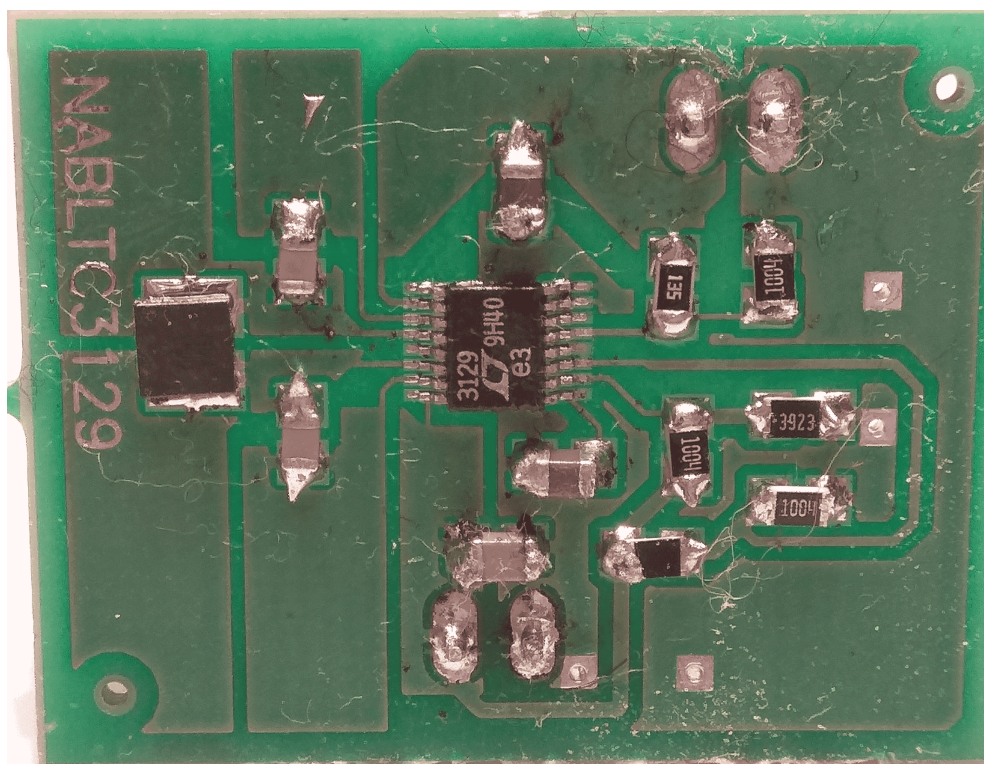
Obvod s LTC3105 po osazení nefungoval. Příčinou bylo nepřipojení pinu SHDN. Po dodatečném připojení tohoto pinu, již obvod fungoval správně, tj. pro různá vstupní napětí v rozsahu 1V - 5V, produkoval na výstupu stabilní výstupní napětí. Osazená deska plošného spoje je na obr. 4.1.



Obr. 4.1: Osazená deska plošného spoje s obvodem LTC3105

4.2 LTC3129

Obvod s LTC3129 po osazení fungoval správně, tzn. pro různý rozsah vstupního napětí produkoval na výstupu stabilní napětí. Výstupní napětí se ale lišilo od napětí požadovaného, tj. 2,8V a určeného pomocí vypočítaných hodnot rezistorů R3, R4. Bylo potřeba poměr těchto rezistorů upravit. Pomohla záměna rezistoru R3 o hodnotě 1,3Mohm za rezistor o hodnotě 1,2M Ω . Osazená deska plošného spoje je na obr. 4.2.



Obr. 4.2: Osazená deska plošného spoje s obvodem LTC3129

4.3 Hlavní deska

Hlavní desku lze rozdělit na několik významných částí, které je potřeba osadit a oživit:

- Napájecí část
- Mikrokontrolér
- IOT modul RN2483
- Snímače
- Paměťová karta
- FTDI pro komunikaci s PC

4.3.1 Napájecí část

Napájecí část desky plošného spoje soustředěná kolem step-up měniče MAX1676 byla osazena a následně proměřena. Napájecí část funguje bez problému, tzn. na výstupu je napětí 3,3V při vstupním napětí od 2V do 3V.

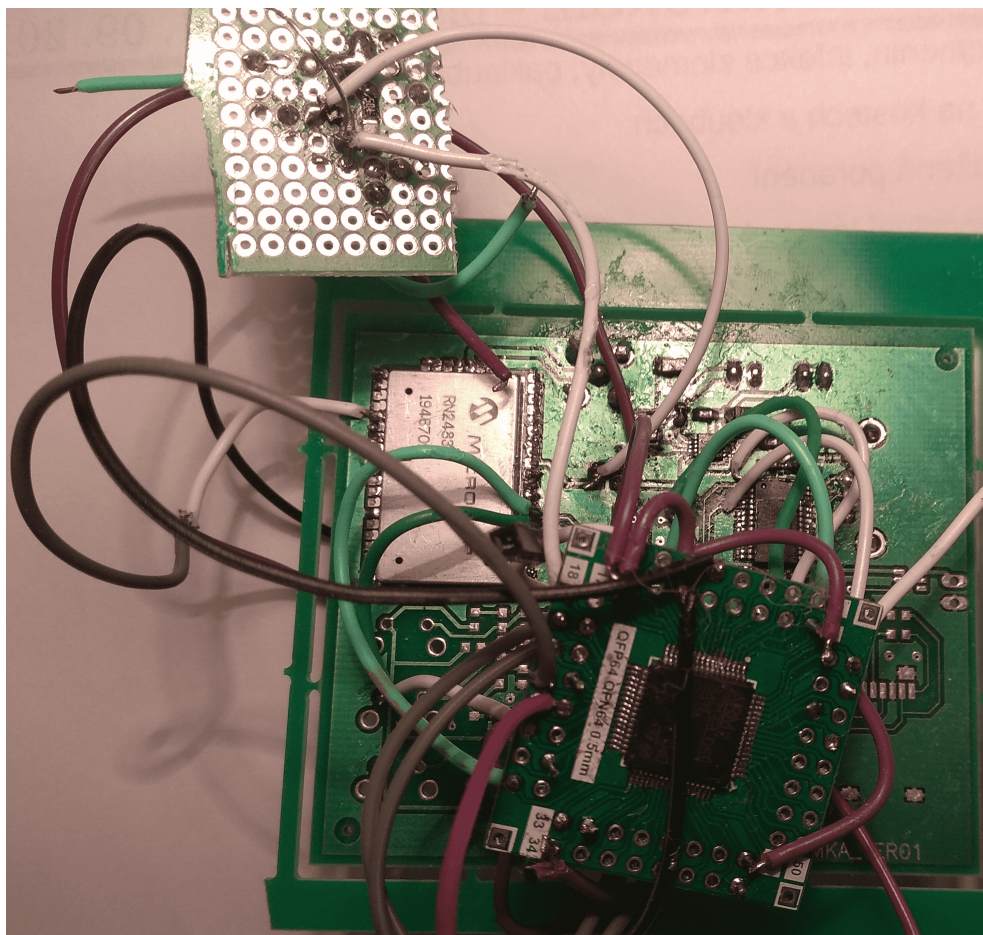
4.3.2 Mikrokontrolér

Bohužel, nejspíše i s přispěním minimální zkušenosti s pájením pouzdra LQFP64, se mi nepovedlo napájet mikrokontrolér na první pokus. Při následném pokusu o odpájení mikrokontroléru, bohužel, došlo k nevratnému poškození plošného spoje v podobě odtržení plošek pro nožičky mikrokontroléru. Nicméně bylo třeba pokračovat v práci, a proto jsem se rozhodl pro nouzové řešení, viz. obr. 4.3. Na univerzální desku plošného spoje byl připájen mikrokontrolér a následně byl tento mikrokontrolér připojen k původní desce plošného spoje pomocí drátů. Bohužel původní typ mikrokontroléru nebyl k dostání v kamenném obchodě, a proto byl dočasně zvolen mikrokontrolér, který se dal koupit. Jedná se o STM32L051. Tento mikrokontrolér má mnohem menší paměť RAM než původní mikrokontrolér, konkrétně 8KB. To má za následek, že není možno implementovat FAT32 systém pro uložení dat na SD kartě. Nicméně k ověření zbývajících funkcí desky plošného spoje a k odstranění případných chyb v zapojení, před odesláním dat pro výrobu další verze, to stačí. Vzhledem ke skutečnosti, že je plánováno využít při implementaci programového řešení knihovny HAL od STMicroelectronics, program by měl být následně snadno portovatelný na původní mikrokontrolér s minimálními změnami.

Po napájení byl k rozhraní SWD připojen Discovery kit, který umožňuje, po minimálních úpravách v podobě odpájení několika nulových rezistorů, přes příslušný konektor programování jiných mikrokontrolérů. Nejprve bylo využito programu STM32 ST-LINK Utility k přečtení flash paměti mikrokontroléru, čímž bylo ověřeno, že je možno s mikrokontrolérem komunikovat. Následně už bylo možno pracovat na programu pro mikrokontrolér v IDE SW4STM32 - System Workbench for STM32.

4.3.3 IOT modul RN2483

Po napájení modulu RN2483 a připojení k mikrokontroléru na univerzální desce pomocí UART bylo přikročeno ke zkoušce komunikace. K této zkoušce byl použit příkaz `sys get vdd`, na který by modul RN2483 měl odpovědět hodnotou svého napájecího napětí. Nicméně modul neodpovídal. Příčinou byla chyba v zapojení, neboť modul neměl připojen pin RESET a vzhledem ke skutečnosti, že má RESET aktivní v logické 0, byl nejspíše celou dobu ve stavu RESET. Nicméně po připojení pinu RESET do stavu logické 1 modul v některém případě odpovídal a v některém případě neodpovídal. Bylo zjištěno, že jediná spolehlivá metoda, jak modul přimět komunikovat, je modul po každé delší chvíli komunikačního klidu resetovat. Proto v další verzi plošného spoje bude možno resetovat modul RN2483 pomocí stavu pinu mikrokontroléru.



Obr. 4.3: Osazení hlavní desky s náhradním mikrokontrolérem na univerzální dps

4.3.4 Snímače

Oba, v této práci popsané, vybrané snímače byly připojeny ke sběrnici I2C a vyzkoušeny. U snímače tlaku BMP280 bylo třeba připojením pinu CSB určit digitální rozhraní a připojením pinu SDO k VCC nebo k GND určit adresu zařízení. Bylo vybráno připojení k VCC, tzn. digitální rozhraní I2C a adresa zařízení je 0x77.

4.3.5 Paměťová karta

Bohužel při návrhu zapojení paměťové micro SD karty došlo k chybě v pinoutu. Nejspíše to bylo zapříčiněno změnou, která byla provedena na poslední chvíli, a to záměnou konektoru SD karty za konektor pro micro SD karty. Tyto karty mají kromě velikosti i jiné rozložení pinů, přičemž pin CS a MOSI byly při této změně opomenuty. Tato skutečnost byla bohužel zjištěna až po napájení konektoru, a proto musel být konektor odpájen a bylo potřeba provést invazivní metodou přerušení spojů k pinům odpovídajícím pinům CS a MOSI původního rozložení a pomocí

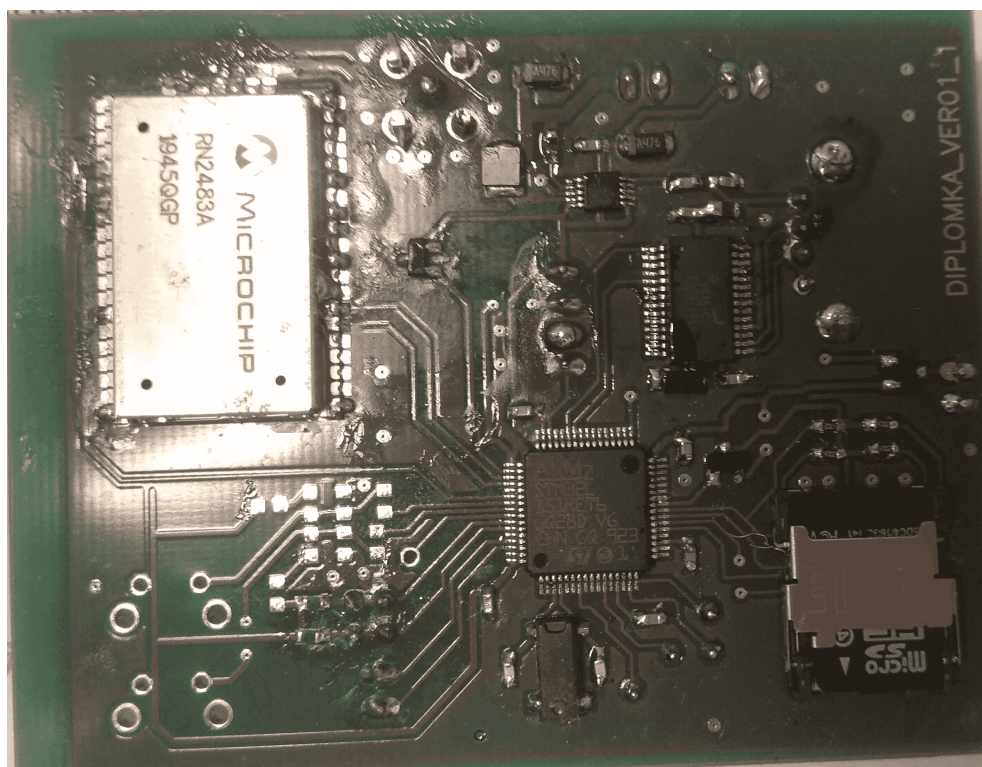
velice tenkého drátku připojit piny CS a MOSI mikrokontroléru ke správným pinům konektoru.

4.3.6 FTDI pro komunikaci s PC

Při tvorbě první verze hlavní desky bylo pro obvod FTDI využito zapojení uvedené v katalogovém listu. Toto schéma ovšem nezohledňuje pin TEST, který bylo potřeba připojit na GND. V jiném případě se zařízení v PC hlásilo jako pokažené. V druhé verzi hlavní desky už tato skutečnost byla zohledněna a zapojení fungovalo na první pokus.

4.3.7 Nová verze hlavní desky

Nová verze hlavní desky byla funkční hned po osazení, přičemž při osazování samozřejmě probíhala kontrola funkčnosti jednotlivých celků. Výsledek je na obr. 4.4.



Obr. 4.4: Osazená 2. verze dps hlavní desky

4.3.8 Krabíčka

Dps s obvodem LTC-3129 byla společně s hlavní deskou, snímači a baterií umístěna do plastové krabíčky, viz. obr. 4.5. Byla vybrána komerčně vyráběná plastová krabíčka, která je bohužel o něco větší než by bylo potřeba. Vhodnější krabíčka ale na trhu nebyla nalezena. Na předním panelu je konektor pro připojení solárního panelu a tlačítko pro zapnutí zařízení. Na zadní straně krabíčky je konektor pro připojení USB kabelu. Jako designové vylepšení do budoucna by bylo možné vyrobit rozměrově optimalizovanou krabíčku na 3D tiskárně. K tomuto kroku nebylo přistoupeno vzhledem k nezkušenostem s 3D tiskem a v době řešení této části nedostupnosti 3D tiskárny k vyzkoušení a naučení.



Obr. 4.5: Zařízení umístěné v krabíčce

5 Programové řešení

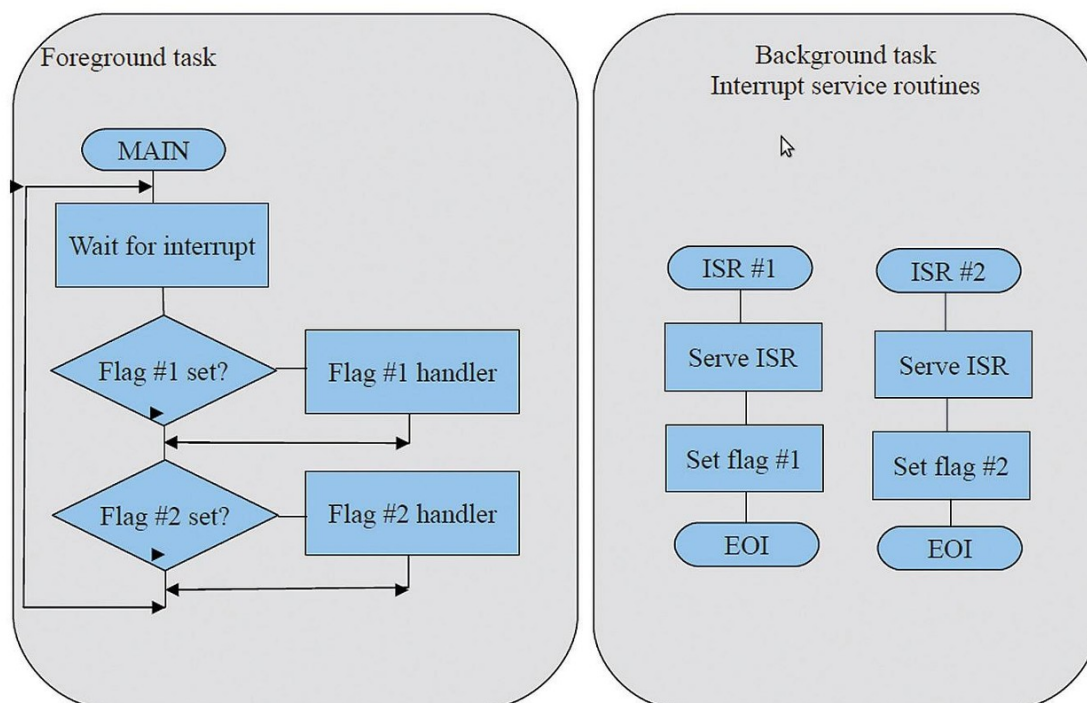
V této části práce je popsáno programové řešení práce, tzn. program pro mikrokontrolér, program pro komunikaci s PC a následně vizualizace naměřených dat.

5.1 Program mikrokontroléru

Projekt byl vygenerován pomocí nástroje STM32CubeMX. Programování probíhalo v IDE System Workbench for STM32(SW4STM32) založeném na prostředí Eclipse. Z důvodu úspory času bylo využito dobře zdokumentovaných knihoven HAL od ST Microelectronics.

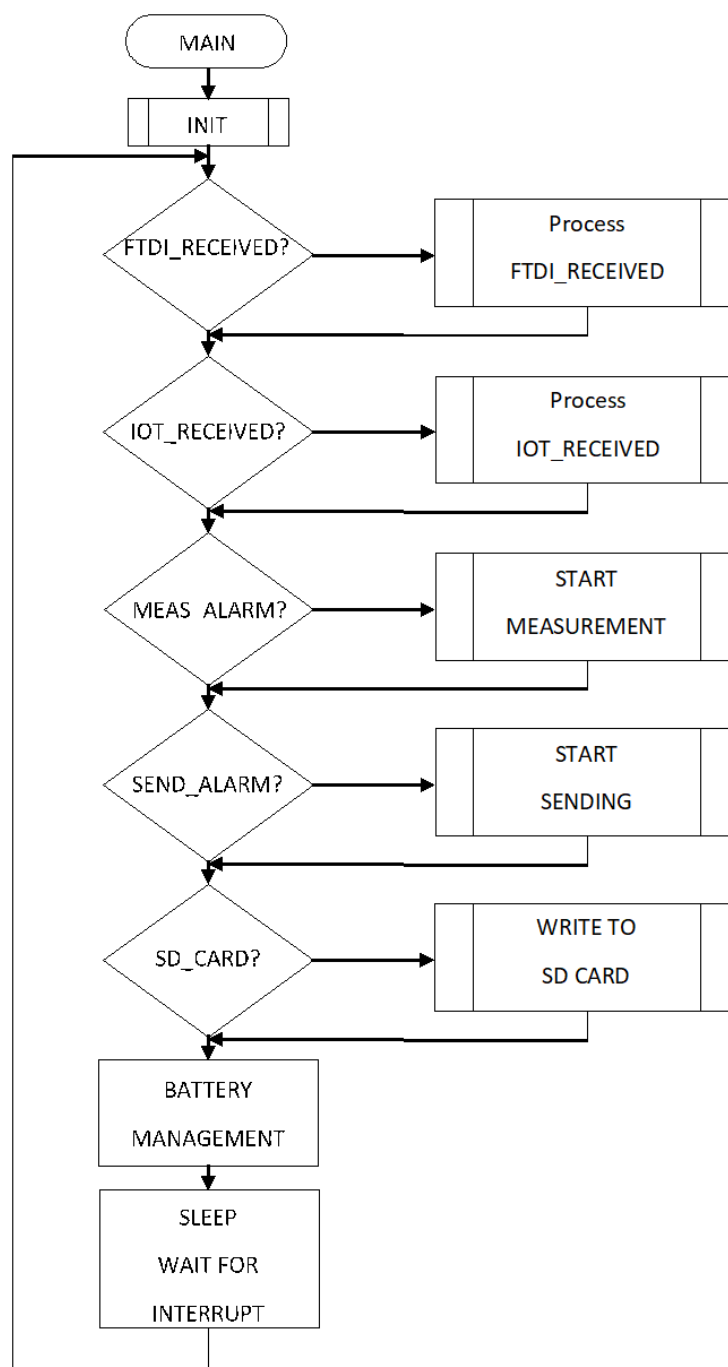
5.1.1 Hlavní smyčka programu

Při vývoji programu byla aplikována technika programování označovaná jako Foreground/Background task. Foreground task volá konkrétní funkce na základě nastaveného bitu globální proměnné, přičemž příslušný bit globální proměnné se nastavuje v Background task, typicky v obsluze přerušení, např. od časovače. Tato technika je též vhodná pro používání různých uspávacích režimů.[18] Pro bližší představu je uveden obr. 5.1.



Obr. 5.1: Technika Foreground/Background Task[18]

V této práci je potřeba obsluhovat snímače, IOT modul, komunikaci s PC a napájecí část. Vývojový diagram pro hlavní smyčku programu je na obr. 5.2¹.



Obr. 5.2: Vývojový diagram hlavní smyčky programu

Na začátku programu proběhne inicializace všech používaných periférií. Následně

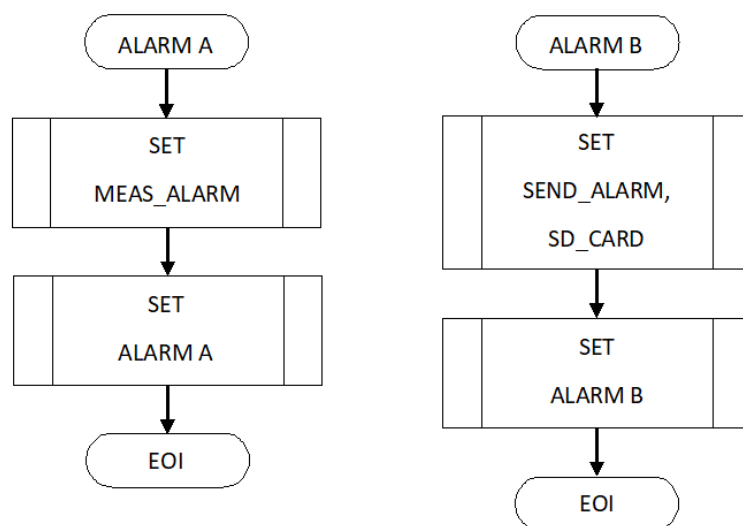
¹U všech vývojových diagramů v této práci se předpokládá, že u rozhodovacího členu ukazuje šipka vpravo na splnění podmínky a šipka dolů na nesplnění podmínky.

v nekonečném cyklu while probíhá testování nastavených flagů pro příslušné obsluhy. Pokud je testování na daný flag pozitivní, je vyvolána příslušná obslužná funkce. Pokud žádný flag není pozitivně testován, je mikrokontrolér uveden do režimu spánku. Z režimu spánku může být mikrokontrolér probuzen jakýmkoliv přerušením, přičemž zde přichází v úvahu:

- Přerušení od alarmu A pro obsluhu snímačů
- Přerušení od alarmu B pro obsluhu IOT modulu
- Přerušení od periferie UART pro obsluhu komunikace s PC

5.1.2 Časované přerušení

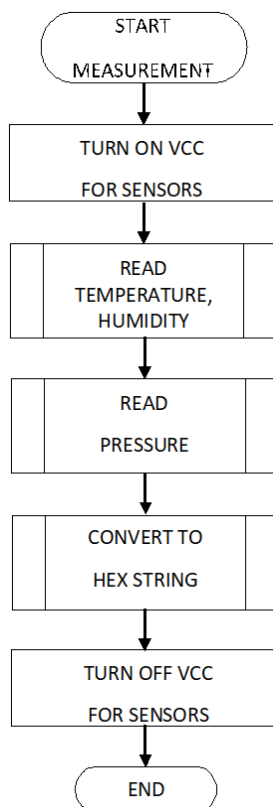
K časovanému přerušení jsou využity 2 RTC alarmy. Alarm slouží k vyvolání přerušení po definovaném čase. Konkrétně je použit z toho důvodu, že přerušení od alarmu probudí mikrokontrolér z režimu spánku a v obslužné rutině přerušení je možno nastavit flag pro vyvolání požadované obslužné funkce. První alarm je použit k nastavení flagu pro měření hodnot ze snímačů. Druhý alarm je použit k odstartování odesílání naměřených hodnot a ukládání hodnot na SD kartu. Po ukončení obslužných funkcí je pomocí AD převodníku změřeno napětí baterie a na základě naměřeného napětí je provedena určitá akce. Pokud je baterie dostatečně nabitá, je ukončeno její dobíjení. Pokud baterie není dostatečně nabitá, je její nabíjení zapnuto nebo není přerušeno. Popsanou funkci znázorňuje vývojový diagram 5.3.



Obr. 5.3: Vývojový diagram časovaného přerušení

5.1.3 Obsluha snímačů

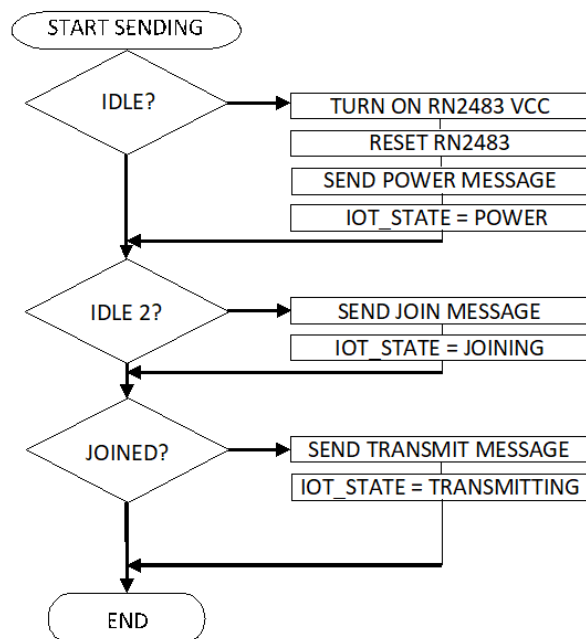
Obsluha snímačů je zaznamenána ve vývojovém diagramu na obr. 5.4. Pokud je nastaven flag pro obsluhu snímačů, přejde se do funkce pro obsluhu. Aktivuje se napájecí napětí pro snímače. Provede se odměření hodnot teploty, vlhkosti, tlaku. Naměřené hodnoty je pro potřeby vysílání hodnot potřeba převést pole datového typu char v hexadecimálním kódu a toto pole připojit do bufferu určeného pro vysílání zprávy. Pokud dojde k zaplnění bufferu určeného pro vysílání zprávy, je buffer smazán a začíná se na začátku.



Obr. 5.4: Vývojový diagram obsluhy snímačů

5.1.4 Obsluha IOT modulu RN2483

Obsluha IOT modulu RN2483 je rozdělena do dvou větví. První větev se, v případě nastavení flagu pro vysílání, stará o vysílání zpráv směrem k modulu RN2483. Druhá větev se, v případě příchozí odpovědi od modulu RN2483, stará o zpracování odpovědi a reakci na odpověď. Popsané chování lze znázornit následujícími 2 vývojovými diagramy 5.5, 5.6.



Obr. 5.5: Vývojový diagram obsluhující vysílání dat

Komunikace s RN2483

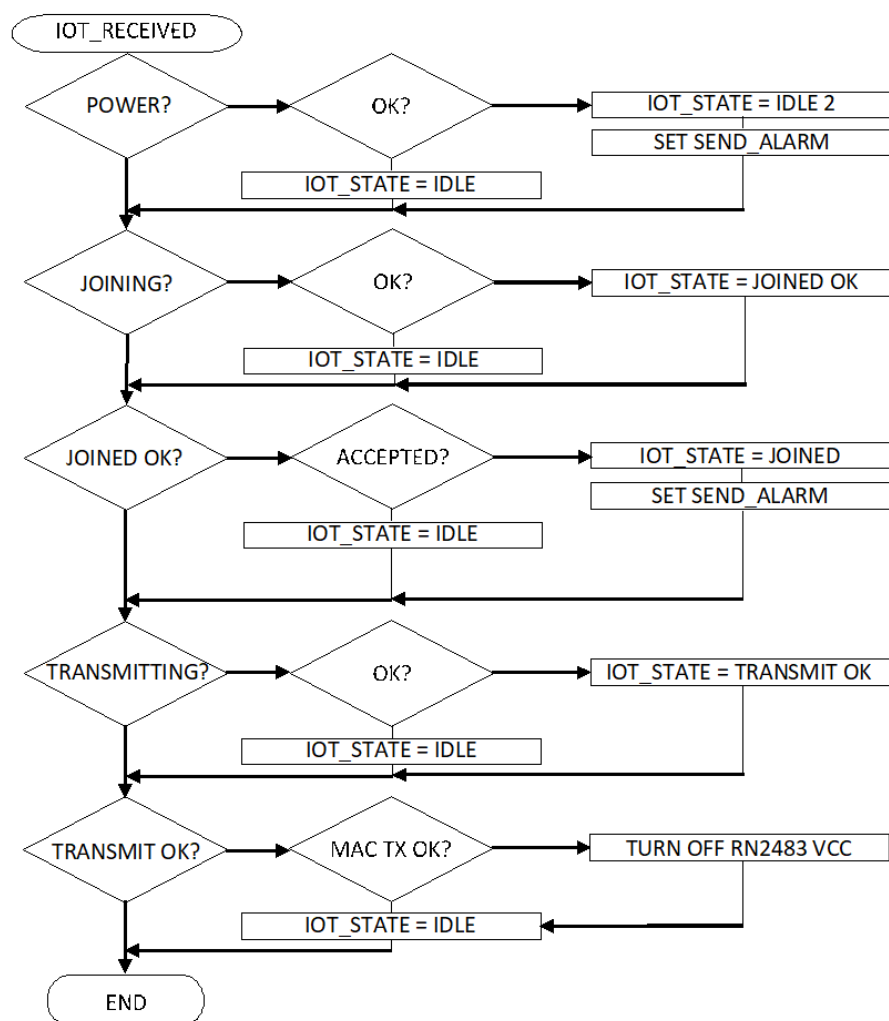
Komunikace s modulem RN2483 probíhá pomocí zpráv, které mají strukturu naznačenou v níže zobrazeném výčtu zpráv. Jde tedy o ASCII zprávy ukončené `\n`. Modul RN2483 podporuje mnoho typů zpráv, všechny jsou popsány v [11]. Pro funkci zařízení požadované v našem případě je potřeba jen několik zpráv. Jsou to zprávy:

- `radio set pwr 14\r\n` - Nastavení síly vysílače - bohužel se musí provést po každém odpojení napájení nebo resetu modulu RN2483. Odpovědí je `ok\r\n`
- `mac join otaa\r\n` - Připojení - odpovědí modulu je `ok\r\n` a následně `accepted\r\n` nebo `denied\r\n`, podle toho zda se povedlo připojit nebo ne
- `mac tx cnf 4 BBCC\r\n` - Odeslání zprávy - BBCC představuje data v hexadecimálním formátu - odpovědí je `ok\r\n`

Struktura zpráv směřujících od IOT modulu RN2483 směrem k mikrokontroléru je stejná jako ve směru opačném.

Obsluha UART

V případě, že je vyvoláno přerušení od periferie uart, zkontroluje se, zda je přijatý byte roven `\n`. Pokud je přijatý byte roven `\n`, je nastaven flag pro obsluhu zpracování odpovědi. Pokud přijatý byte není roven `\n`, je přijatý byte zařazen do bufferu. Popsaná funkce odpovídá vývojovému diagramu 5.7.



Obr. 5.6: Vývojový diagram obsluhující příjem dat z modulu RN2483

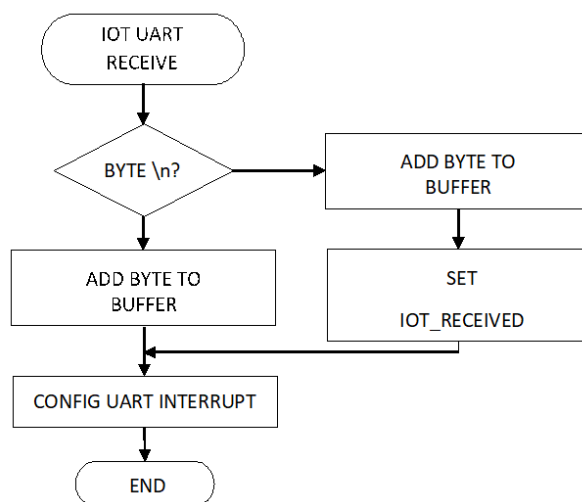
Odesílání zpráv probíhá pomocí DMA. Z důvodu jednoduchosti nebude uveden vývojový diagram pro odesílání dat z mikrokontroléru k IOT modulu RN2483.

5.1.5 Obsluha komunikace s PC

V případě nastavení flagu pro komunikaci s PC, je zavolána příslušná obslužná funkce. Následně dochází k dekódování zprávy a patřičné reakci, tzn. odpovědi v podobě aktuálních hodnot. Situaci přehledně popisuje vývojový diagram 5.8.

5.1.6 Komunikační protokol pro komunikaci s PC

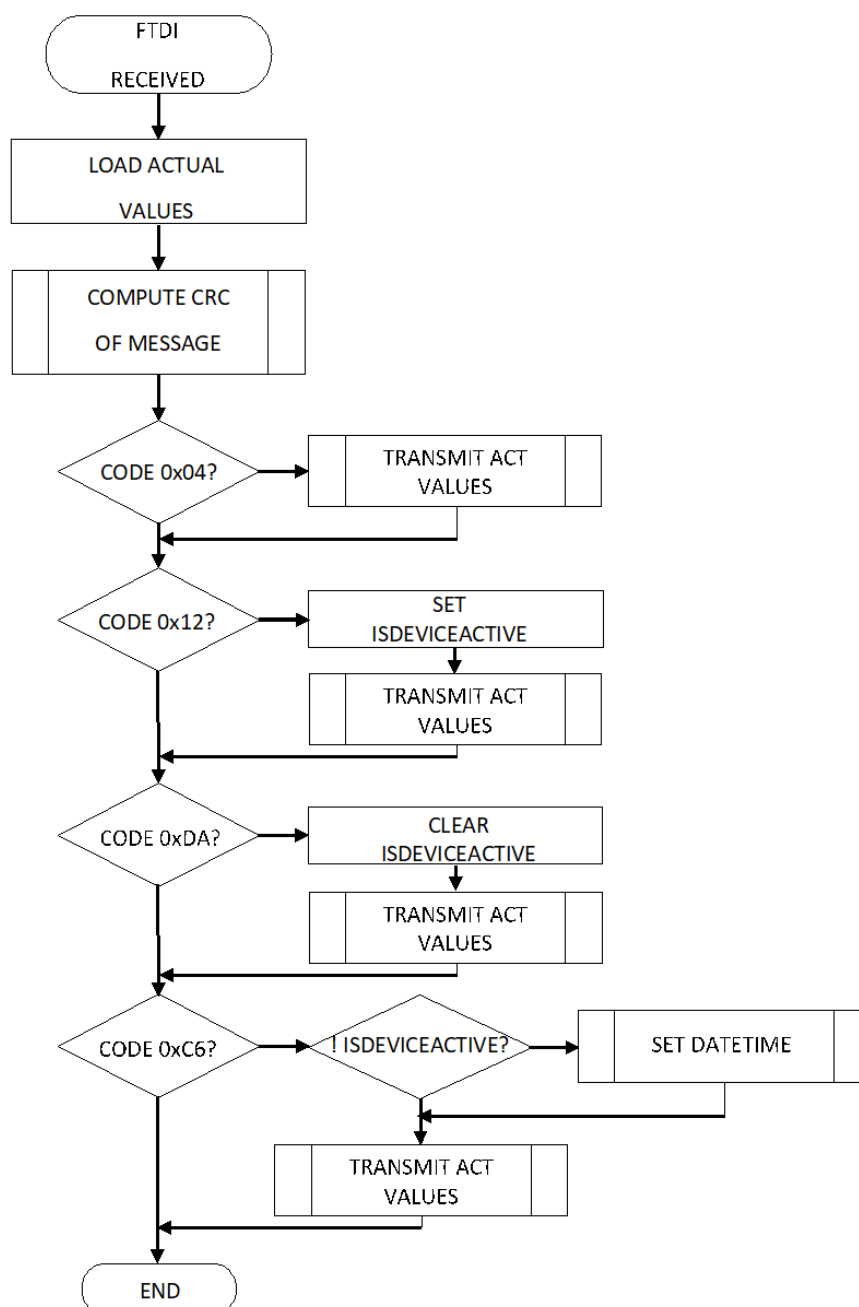
Protože je potřeba pouze jednoduchých pokynů od PC směrem k zařízení, nebyl implementován některý ze standardních komunikačních protokolů, nýbrž byl navržen vlastní jednoduchý komunikační protokol s inspirací u protokolu modbus. Protokol je



Obr. 5.7: Vývojový diagram obsluhy příjmu dat od RN2483 na uart

založen na principu master(PC) - slave(Embedded zařízení). Iniciátorem komunikace může být pouze master. Struktura zprávy od mastera je zobrazena na obr. 5.9. Struktura odpovědi od zařízení typu slave je na všechny zprávy stejná a její struktura je zachycena na obr. 5.10. Zprávy mají konstantní délku. Délka zprávy od zařízení typu master je vždy stejná. To platí i pro zprávu od zařízení typu slave. Nicméně délka zprávy od zařízení typu master je jiná oproti délce zprávy od zařízení typu slave. Díky známé délce podporovaných zpráv není potřeba ve zprávě ukládat počet přenášených bytů jako např. u protokolu modbus nebo řešit ukončovací znak. 13 Bytů dat, která přenáší zařízení typu slave směrem k PC reprezentuje následující výčet. Pořadí ve výčtu odpovídá pořadí v rámci zprávy.

1. Měření běží - 0x55, Měření nebylo spuštěno - 0
2. Teplota - horní byte
3. Teplota - spodní byte
4. Vlhkost - horní byte
5. Vlhkost - spodní byte
6. Tlak - horní byte
7. Tlak - spodní byte
8. Datum - den
9. Datum - měsíc
10. Datum - rok
11. Čas - hodina
12. Čas - minuta
13. Čas - sekunda



Obr. 5.8: Vývojový diagram obsluhy komunikace s PC

0x7E	0x05	Kód fce	6BYTES DATA	CRC	CRC
------	------	---------	-------------	-----	-----

Obr. 5.9: Struktura zprávy od PC směrem k slave zařízení

Podporované zprávy od PC směrem k zařízení

Podporované zprávy od PC směrem k zařízení typu slave uvádí následující výčet:

0x7E	13BYTES DATA	CRC	CRC
------	--------------	-----	-----

Obr. 5.10: Struktura zprávy od slave zařízení směrem k PC

- Pošli aktuální údaje - 0x04
- Odstartuj měření - 0x12
- Zastav měření - 0xDA
- Nastav datum a čas - 0xC6
- Přejdi do režimu spánku - 0x4D

Struktura zprávy typu Nastav datum a čas je na obr. 5.11.

0x7E	0x05	0xC6	DD	MM	YY	HH	mm	SS	CRC	CRC
------	------	------	----	----	----	----	----	----	-----	-----

Obr. 5.11: Struktura zprávy typu Nastav datum a čas

Obsluha UART

Příchozí zpráva vyvolá přerušení od periférie UART. Dochází ke kontrole zprávy, tzn. kontroluje se, zda první 2 byty odpovídají správným hodnotám a byty náležící kontrolnímu kódu CRC odpovídají vypočítanému CRC zprávy. Pokud je zpráva v pořádku, je zkopírována do bufferu a nastaven flag pro obsluhu komunikace s PC. Situaci znázorňuje vývojový diagram 5.12.

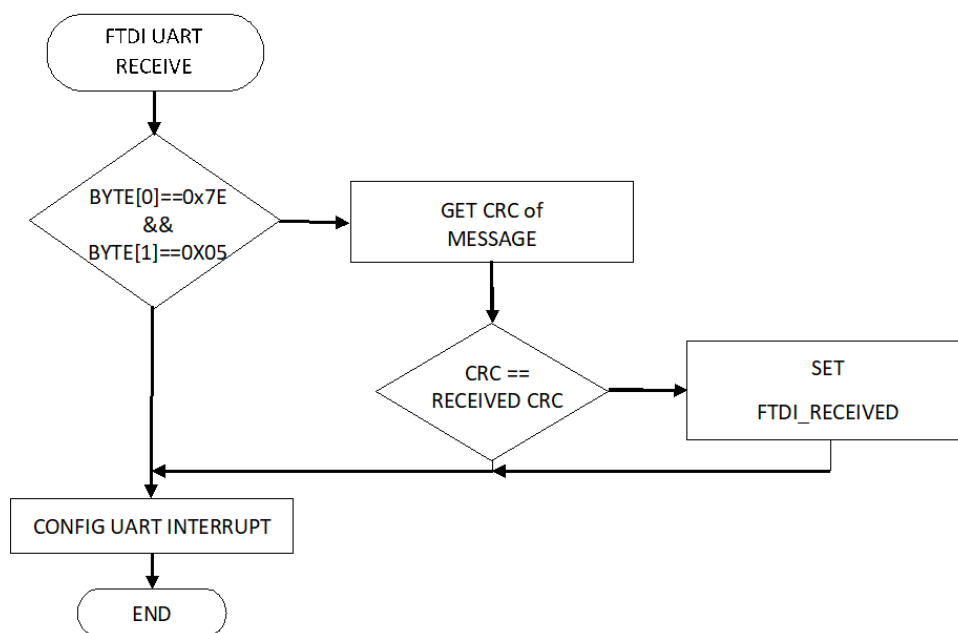
V případě odchozí zprávy je zpráva poskládána do pole, dle výše popsané struktury zprávy. Vypočte se kontrolní kód CRC, zařadí se na konec zprávy a zpráva je odeslána pomocí DMA.

5.1.7 Ukládání dat na SD kartu


Naměřená data jsou ukládána do textového souboru. Název textového souboru je dán časovou značkou spuštění měření. Časová značka je ve tvaru DD/MM/YY-/HH/mm/ss, příklad názvu souboru tedy může být 120120142532.txt. Struktura souboru obsahuje následující data:

- Časová značka
- Teplota
- Vlhkost
- Tlak

Ukázka záznamu dat je na obr. 5.13. Formát dat teploty, vlhkosti a tlaku odpovídá 100× vynásobené hodnotě dané veličiny převedené do hexadecimálního formátu.



Obr. 5.12: Vývojový diagram pro obsluhu uart od obvodu ftdi

 090420195800 – Poznámkový blok

Soubor	Úpravy	Formát	Zobrazení	Nápověda	
04-	9-20	19:59:01	0997	0FA7	0000
04-	9-20	20:00:01	0000	2710	0000
04-	9-20	20:01:01	0996	0FBE	0000
04-	9-20	20:02:01	0995	0FCC	0000
04-	9-20	20:03:01	0000	2710	0000
04-	9-20	20:04:01	0000	2710	0000
04-	9-20	20:05:01	0000	2710	0000
04-	9-20	20:06:01	0000	2710	0000

Obr. 5.13: Ukázka textového souboru se záznamem dat

FAT systém

Byl využit FatFs modul [16]. K tomuto modulu je potřeba připojit nejnižší vrstvu, tzn. konkrétní driver, k čemuž byl využit driver poskytovaný z [17].

5.1.8 Optimalizace na nízký odběr energie

Optimalizace zařízení na nízký odběr začala již při návrhu zapojení. Konkrétně tím způsobem, že lze odepnout napájení snímačům, paměťové kartě i IOT modulu. Pro další optimalizaci je využito následujících technik:

- Clock gating
- Analog mód pro nevyužité piny
- Low Power módy mikrokontroléru

Clock gating

Clock gating je jednoduše řešeno vypnutím hodin(Clock) perifériím, které nejsou potřeba. Tato technika byla využita ve 2 případech:

1. Vypnutí hodin perifériím, které nejsou potřeba v Run módu
2. Vypnutí hodin před vstupem do Sleep módu u periférií, které nejsou ve Sleep módu potřeba. Po probuzení z režimu spánku je potřeba hodiny opětovně zapnout

Analog mód pro nevyužité piny

Nevyužité piny mikrokontroléru lze nastavit do analog módu. To je výhodné z hlediska energetické náročnosti vzhledem k následujícím vlastnostem tohoto módu:

- Výstupní buffer je zakázán
- Vstup Schmittova klopného obvodu je deaktivován, což zajišťuje nulový odběr pro analogové hodnoty na daných pinech mikrokontroléru. Výstup Schmittova klopného obvodu je 0.
- Pull-up a Pull-down rezistory jsou hardwarově zakázány

Low Power módy mikrokontroléru

U mikrokontroléru STM32L151 je možné využít několik Low Power módů:

- Sleep mód
- Low-power Run mód
- Low-power Sleep mód
- Stop mód s RTC
- Stop mód bez RTC
- Standby mód s RTC
- Standby mód bez RTC

Požadavky na Low Power mód vzhledem k aplikaci jsou shrnuty v následujícím seznamu:

- Zachování obsahu paměti RAM

- Možnost probuzení pomocí přerušení RTC Alarmu
- Možnost probuzení pomocí UART periferie

Po srovnání výše zmíněných požadavků a popisu Low Power módů v [14] byl vybrán Low Power sleep mód. Z tohoto módu bohužel nelze mikrokontrolér probudit periferií UART. Tento problém byl vyřešen následujícím způsobem: Před uspaním mikrokontroléru je periferie UART odinicializována a pro pin odpovídající pinu RX Uart je nastaveno externí přerušení. Pokud PC vyšle zprávu směrem k mikrokontroléru, vyvolá se externí přerušení a v příslušné funkci se provede inicializace periferie UART. Následně již probíhá komunikace klasickým způsobem. Nutno poznamenat, že tímto způsobem dojde ke ztrátě první vyslané zprávy, což ale v tomto případě nevádí.

Dosažené výsledky

Po implementaci 3 výše popsaných technik určených pro optimalizaci proudového odběru bylo dosaženo proudového odběru v nečinnosti 1.2 mA. Oproti teoretické hodnotě určené při zpracování energetické bilance je tato hodnota mnohem vyšší. Pokud v tab. 2.3 v řádku pro Režim spánku a sloupci Odběr zaměníme hodnotu 0.02 mA za hodnotu 1.52 mA, dojdeme k výsledku průměrně odebíraného proudu cca $5\times$ vyššímu, viz. následující výpočty.

$$I_c = I_{mer} + I_{Vys} + I_{zap} + I_{spanek} = 0.0152 + 0.12 + 0.08 + 0.8892 = 1.1044A/10min \quad (5.1)$$

$$I_{avg} = \frac{I_c}{t} = \frac{1.1044}{600} = 1.84mA \quad (5.2)$$

Následně vydělíme skutečný průměrný proudový odběr s teoretickým průměrným proudovým odběrem.

$$I_{pomer} = \frac{I_{avg}}{I_{avgteor}} = \frac{1.84}{0.37} = 5 \quad (5.3)$$

Ostatní hodnoty v tabulce měnit nemusíme, neboť se ve výpočtu projeví minimálně.

Hlavním důvodem vyššího proudového odběru oproti teoretickému je nízká účinnost step-up měniče. Bohužel nelze změřit odebíraný proud za měničem na nové verzi hlavní desky. Proto by bylo potřeba vytvoření měřícího bodu již při návrhu desky, např. pomocí jumperu na hlavní napájecí cestě za měničem a při jeho rozpojení by bylo možné do cesty vložit např. ampérmetr. S tímto se však při návrhu nepočítalo. Nicméně na první verzi hlavní desky s jiným mikrokontrolérem byla přerušena hlavní napájecí cesta a byl takto vytvořen měřící bod. Na první verzi hlavní desky bylo měřením ověřeno, že step-up měnič má účinnost v režimu nečinnosti cca 10procentní, přičemž podle datasheetu [15] měla být účinnost cca 80%.

U step-up měničů někdy bývá problém s vysokým sériovým odporem (ESR) kondenzátorů. Při výběru součástek byla snaha vybrat kondenzátory s co nejnižším ESR. Na radu vedoucího práce bylo vyzkoušeno nahradit větší kondenzátory dvěma menšími, nicméně to nepřineslo požadovaný výsledek v podobě zvýšení účinnosti. Další možnost je, že je nevhodně navržený layout dps u step-up měniče, nicméně toto nelze ověřit bez tvorby nové dps.

5.2 Program pro ovládání zařízení z PC

V této části práce je popsán program vytvořený pro ovládání zařízení z PC. S FTDI obvodem FT232RL použitým v této práci je možno se zařízením připojeným pomocí USB, po instalaci příslušných ovladačů, pracovat jako s virtuálním COM portem. Pro práci s COM portem existuje samozřejmě mnoho knihoven a program se dá vytvořit v mnoha programovacích jazycích, např. v oblíbeném pythonu. Já jsem se rozhodl program vytvořit ve vysokoúrovňovém objektově orientovaném jazyce C#, a to hned ze dvou důvodů. Prvním důvodem je vlastní pozitivní zkušenost s tímto jazykem a druhým důvodem je, že bude potřeba vytvořit GUI aplikaci a společnost Microsoft nabízí vývojářům několik možností, jak vytvořit GUI aplikaci v jazyce C#, a tyto možnosti jsou určitým způsobem integrované v nástrojích Microsoft Visual Studio, což vývojářům značně ušetří práci a čas.

Dříve bylo standardem vytvářet grafické aplikace v C# společně s frameworkem WinForms. Dnes je tento framework již zastaralý a byl nahrazen novějším frameworkem WPF - Windows Presentation Foundation. V této práci bude využit framework WPF.

Aby bylo aplikaci možno jednoduše rozšiřovat, opravovat, atd., je potřeba vyvarovat se tzv. špagetového kódu a tzv. code behind. Tomuto požadavku se nejsnáze vyhoví následováním některého z návrhových vzorů, který důsledně odděluje grafickou část zobrazovanou uživateli a část starající se o logiku aplikace. Společnost Microsoft doporučuje pro psaní formulářových aplikací s pomocí frameworku WPF dodržovat design pattern MVVM - Model, View, View Model. Program pro ovládání zařízení z PC bude tedy psán podle design patternu MVVM. Požadavky na program jsou shrnuty v následujícím seznamu:

- Práce s virtuálním COM Portem
- Implementace v této práci popsaného komunikačního protokolu (Master)
- Funkce:
 - Konfigurace zařízení
 - Spuštění měření
 - Zastavení měření
 - Sledování posledních naměřených hodnot

- Grafické uživatelské rozhraní
- Následování návrhového vzoru MVVM

5.2.1 Návrhový vzor MVVM

MVVM vycházející z Model-View-ViewModel je návrhový vzor pro oddělení logiky od uživatelského rozhraní při programování WPF aplikací. MVVM odděluje data, stav aplikace a uživatelské rozhraní. Návrhový vzor tedy obsahuje 3 hlavní komponenty:

- Model
- View
- ViewModel

Model

Data se kterými aplikace pracuje jsou popsány pomocí Modelu. Model může pracovat s daty z databáze nebo v případě aplikace navrhované v této práci bude model představovat práci s daty COM Portu.[19]

View

View představuje uživatelské rozhraní definované v jazyce XAML.[19]

ViewModel

Spojení Modelu a View má na starosti ViewModel. ViewModel si drží stav aplikace, přičemž ovládací prvky aplikace jsou pomocí Bindingu propojeny s ViewModelem.[19]

Princip

Třída typu ViewModel si drží stav aplikace. Podle této třídy uživatelské rozhraní vykresluje uživatelské prvky. Změní-li uživatel v uživatelském rozhraní nějaká data, tato data se vypropagují do ViewModelu - k tomu je určen Binding. Naopak, změnil se některá z vlastností ve ViewModelu je vyvolána událost, která informuje View o změně a dojde k automatickému překreslení.[19]

5.2.2 Model - Obsluha COM Portu

Model v našem případě představuje práce s COM portem. Základem modelu je třída StmService. Třída StmService ve svém konstruktoru vytváří instanci třídy SerialPort. Tato třída umožňuje nastavovat následující parametry komunikace:

- Jméno portu

- Baudrate
- Parita
- Počet datových bitů
- Počet stopbitů

Třída dále obsahuje proměnné(atributy) sledovaných hodnot - teplota, vlhkost, tlak a atributy spojené s konkrétním spojením, časový interval spojení, stav spojení - připojeno, odpojeno, připojuje se. Třída `StmService` dále obsahuje časovač třídy `System.Timers.Timer`, který periodicky vyvolává událost přečtení aktuálních hodnot z embedded zařízení - v podobě zapsání dat na port. Třída `StmService` dále obsahuje metody `StartMeasurement` a `StopMeasurement`, které jsou, jak již z názvu vypovídá, určené ke spuštění nebo zastavení měření. Další metodou je `SetDateTime` pro nastavení času embedded zařízení. Struktura odesílaných zpráv odpovídá výše popsanému komunikačnímu protokolu.

Po zapsání dat na port se čeká na odpověď zařízení, konkrétně je k tomu určena metoda `GetResponse`, která má nastavený timeout, kdy čeká, zda se na portu objeví data. Pokud do vypršení timeoutu nejsou data k dispozici, vrací metoda neúspěšnou odpověď. Pokud se data před vypršením timeoutu objeví, přečte data a pomocí kontrolního kódu CRC ověří jejich správnost. V případě úspěšného ověření vrací metoda úspěšnou odpověď a dojde k aktualizaci sledovaných hodnot. Diagram třídy `StmService` je zobrazen na obr. 5.14 vlevo.

5.2.3 ViewModel

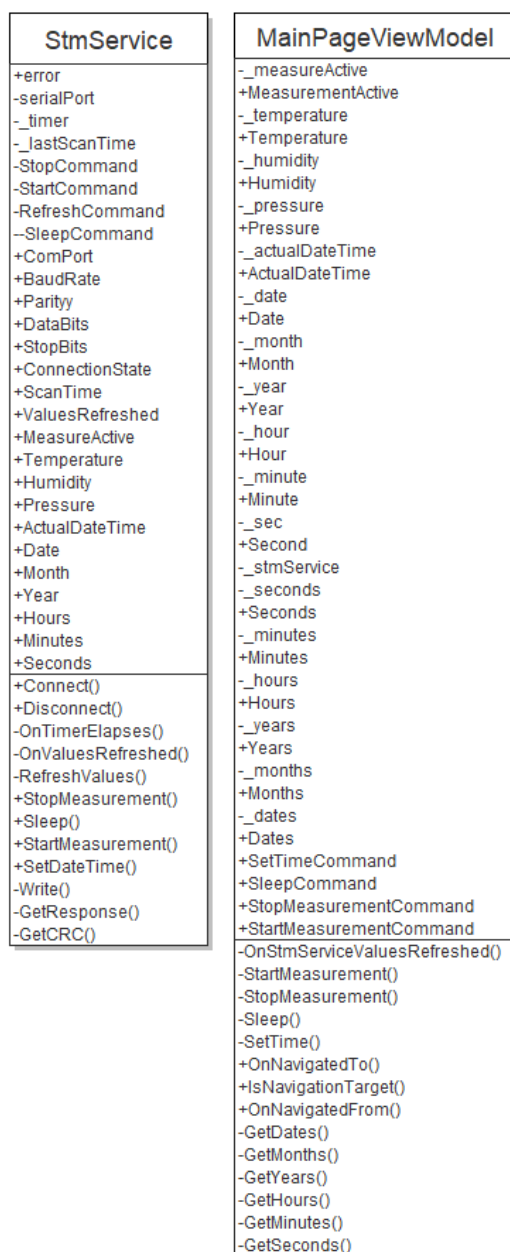
Aplikace obsahuje 5 ViewModels. Každému ViewModel odpovídá příslušné View. ViewModels jmenovitě jsou:

- `LeftMenuViewModel` - stará se v podstatě jen o navigaci mezi hlavní stránkou a stránkou s nastavením parametrů komunikace
- `MainPageViewModel` - pomocí tzv. dependency injection je jí k dispozici instance třídy `StmService`, monitoruje sledované parametry a stará se o vykonání příkazů přijatých z Views. Diagram třídy je na obr. 5.14.
- `MainWindowViewModel` - slouží pouze jako manažer pro registraci příslušných regionů
- `SettingsPageViewModel` - stará se pouze o nastavení parametrů komunikace
- `StmStatusBarViewModel` - slouží pouze pro monitorování stavu připojení

Diagramy tříd `LeftMenuViewModel`, `MainWindowViewModel`, `SettingsPageViewModel`, `StmStatusBarViewModel` jsou na obr. 5.15.

5.2.4 Views

Každému z výše zmíněných ViewModels náleží příslušné View:

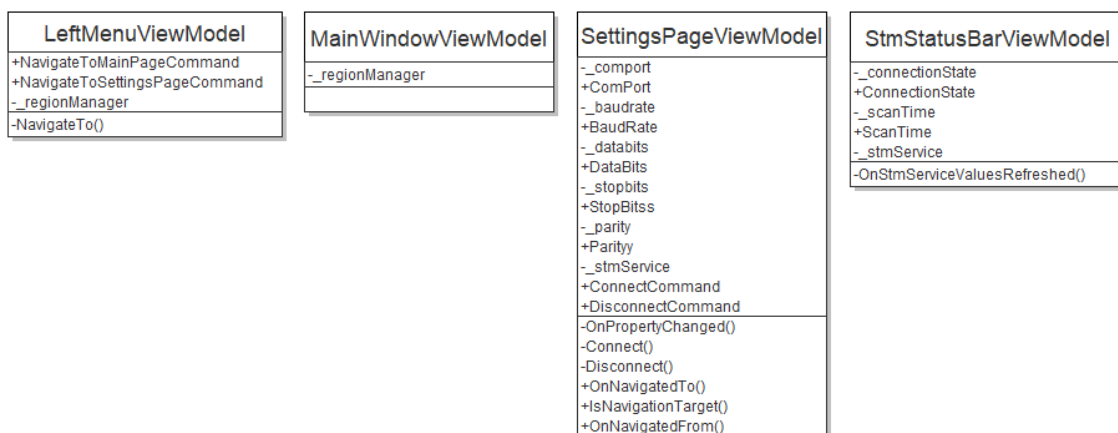


Obr. 5.14: Diagram tříd StmService(vlevo) a MainPageViewModel(vpravo)

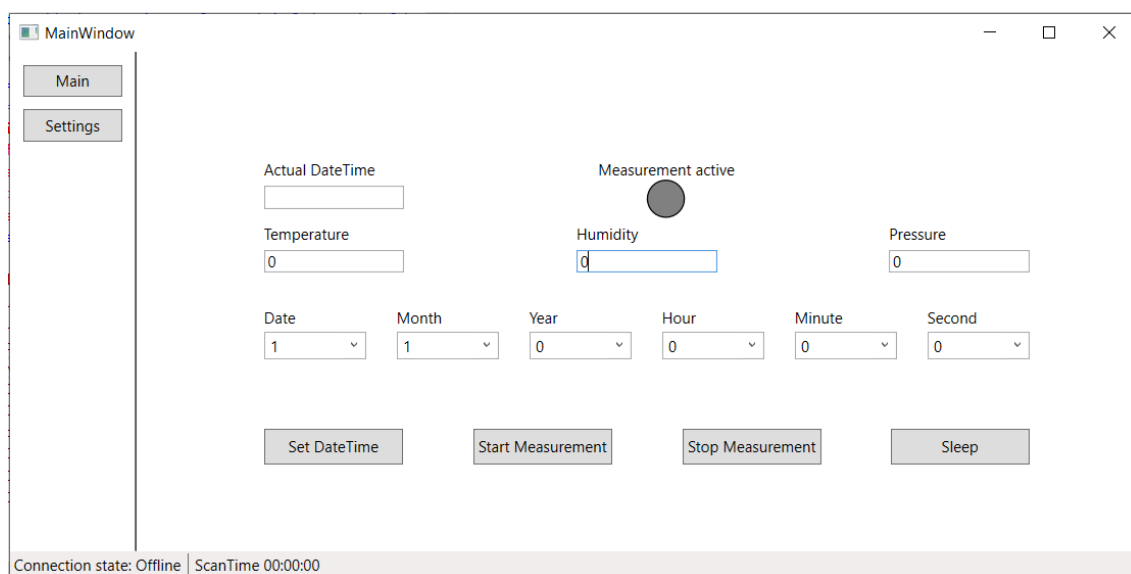
- LeftMenu
- MainPage
- MainWindow
- SettingsPage
- StmStatusBar

Na obr. 5.16 je ukázána hlavní stránka aplikace.

Hodnota Measurement active je reprezentována barvou - zelená barva symbolizuje logickou hodnotu true a šedá barva symbolizuje logickou hodnotu false. Ke



Obr. 5.15: Diagram tříd náležících ViewModels



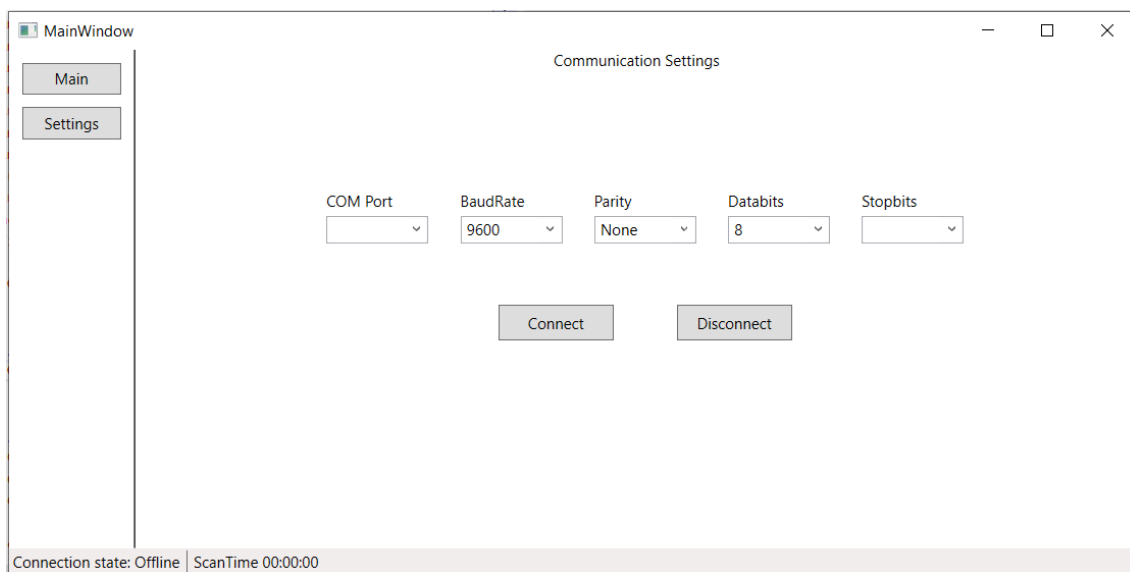
Obr. 5.16: Ukázka hlavní stránky aplikace

konverzi z logické hodnoty na vyjádření barvou je potřeba použít mimo výše zmiňované kategorie(Model, View, ViewModel) tzv. Converter. Zde je tedy konkrétně použit BooleanToBrushConverter.cs, konvertující z logické hodnoty na barvu. Diagram třídy je na obr. 5.17.



Obr. 5.17: Diagram třídy BooleanToBrushConverter

Na obr. 5.18 je ukázána stránka s nastavením parametrů komunikace.



Obr. 5.18: Ukázka stránky pro nastavení parametrů komunikace

5.3 Návod na konfiguraci zařízení z PC

V této části práce je popsáno jak nakonfigurovat zařízení a spustit měření. Jednotlivé kroky je nutno provádět v pořadí ve kterém jsou napsány.

1. Propojte zařízení s PC pomocí USB kabelu
2. Spusťte aplikaci určenou pro PC
3. Přejděte na záložku Settings
4. Nastavte parametry komunikace s následujícími hodnotami
 - COM Port - port odpovídající připojenému zařízení
 - BaudRate - 38400
 - Parity - None
 - DataBits - 8
 - StopBits - 1
5. Stiskněte tlačítko Connect
6. Přejděte na záložku Main
7. Nastavte časovou značku
8. Stiskněte StartMeasurement pro spuštění měření - ikona Measurement active bude svítit zeleně
9. Stiskněte Sleep pro uspání zařízení a ukončení komunikace
10. Odpojte zařízení od PC

Pokus nestisknete tlačítko Sleep, nepřejde mikrokontrolér do režimu spánku a bude nadále komunikovat s PC, tzn. že po uběhnutí příslušného časového limitu se budou měnit naměřené hodnoty teploty, vlhkosti a tlaku. Též lze vidět aktuální časovou značku.

5.4 Vizualizace měřených dat

V této části práce je popsána vizualizace naměřených dat, včetně všech mezikroků vedoucích k samotné vizualizaci.

5.4.1 Služby CRA

Ze strany Českých radiokomunikací mi bylo umožněno využívat jejich IOT služeb za účelem testování zdarma. Byl mi tedy zřízen účet v IOT CRA portálu, který je určen pro správu zařízení.

Následně bylo potřeba zaregistrovat své zařízení. K tomu je potřeba nastavit následující parametry:

- Identifikátor zařízení DevEUI
- Device application EUI AppEUI
- Aplikační klíč AppKey

Výše zmíněné parametry je potřeba mít stejně nastavené na modulu RN2483. To je možno provést v zásadě 2 způsoby:

- Připojit se k pinům patřícím uart modulu RN2483 převodníkem USB/UART, např. CP2102, a následně se připojit k příslušnému COM Portu pomocí některého z dostupných terminálů, např. TeraTerm. Poté pomocí příkazů z dokumentace modulu RN2483 nastavit příslušné parametry
- Využít mikrokontrolér, který je připojený k uart modulu RN2483 a napsat jednoduchý program, který nastaví příslušné parametry

Po registraci je Vám umožněno sledovat poslední aktivitu svého zařízení, sílu signálu, viz. obr. 5.19. Dále je možno stáhnout si zprávy odeslané Vaším zařízením ve formátu CSV.

The screenshot shows the CRA IoT Portal interface. On the left is a dark sidebar with the CRA IoT Portal logo and a menu with options: Účty, Skupiny zařízení, Endpointy, Služby, Import zařízení, Zařízení (highlighted), and Ověření pokrytí. The main area has a header with 'Účet' (Petr Raichl) and 'Zákazník' (Na zkoušku CRA IoT), and a user profile for 'Petr Raichl' with an 'ODHLÁSIT' button. Below the header, the 'Zařízení' section shows a 'Filtrování dat' link and a table with one device entry.

DevEUI	Název zařízení	Poslední aktivita	Síla signálu	Akce
0004A30B00EF4957	RN2483_1	-15 sec	<div></div>	

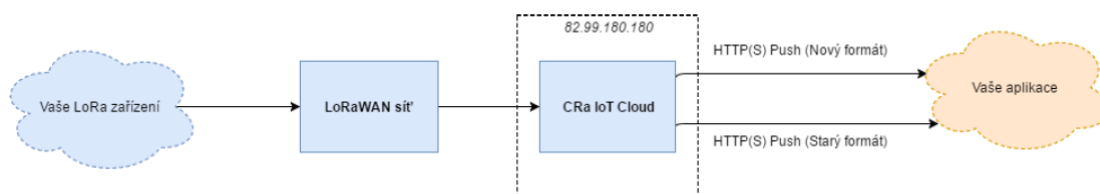
Obr. 5.19: Ukázka CRA IOT portálu

Problémy

Pokud se podíváte na mapu pokrytí dostupnou na <https://www.cra.cz/sluzby-iot>, zdá se, že je většina České republiky pokrytá. Detailní ověření pokrytí máte až po zřízení přístupu do IOT portálu. Bohužel, v místě odkud jsem standardně pracoval na této práci signál uvnitř domu ani venku nebyl, což práci poněkud komplikovalo.

CRA Vizualizace dat

CRA IOT portál bohužel neumožňuje vizualizaci dat, ale je možno nastavit přeposílání zpráv ze strany CRA na Váš endpoint. Tuto situaci vystihuje obr. 5.20.



Obr. 5.20: Tok dat od zařízení k aplikaci pro vizualizaci [20]

Konkrétně jsou data odesílána ve formátu JSON, který obsahuje veškeré informace o odeslané zprávě z Vašeho zařízení. Mezi nejdůležitější patří:

- DevUI
- Data
- čas

Další informace mají spíše doplňující charakter. Mezi tyto informace patří například síla signálu, počet stanic, které přijali zprávu.

Ukázka přeposílané zprávy ve formátu JSON je na obr. 5.21.

5.4.2 AWS

Pro vytvoření endpointu na které by se přeposílali zprávy z portálu Českých radiokomunikací by šlo využít mnoha IOT služeb, z nichž některé jsou poskytovány s určitými podmínkami zdarma. Z těchto služeb lze zmínit například Thingspeak, atd. Z komplexních služeb, vhodných i pro větší komerční projekty, lze zmínit Azure IOT od společnosti Microsoft, Google Cloud IOT nebo AWS IOT od Amazonu.

Já jsem se rozhodl vytvořit své vlastní řešení. A to v následující podobě shrnuté v bodech:

- Data z CRA budou přeposílány do vlastní databáze
- Vizualizace dat z databáze bude probíhat ve webové aplikaci naprogramované v ASP.NET Core

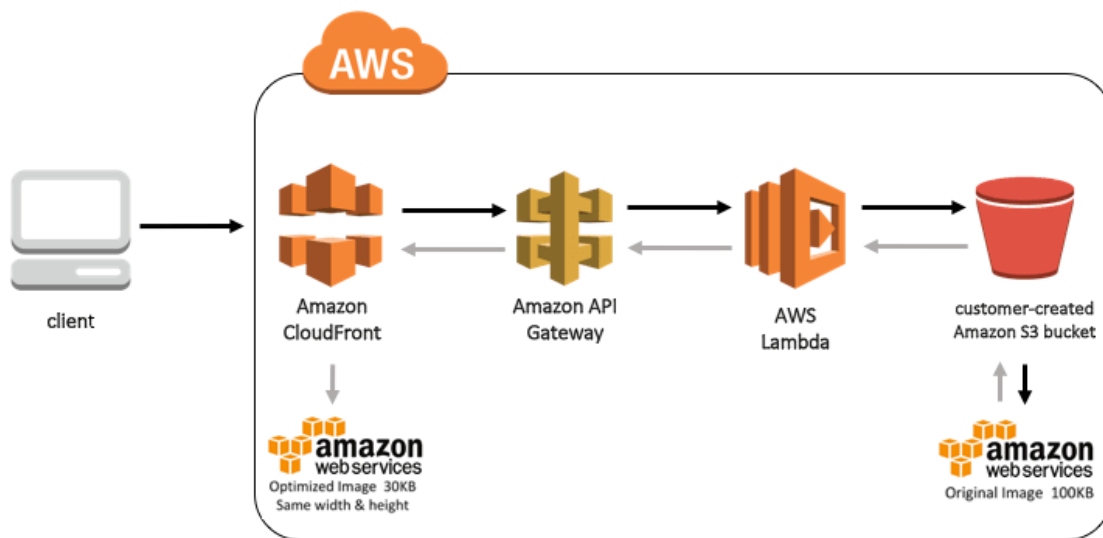
Content-Type: application/json

```
{"type":"D","data":{"cmd\\":\\"gw\\",\\"EUI\\":\\"XXXXXXXXXXXXXXXX\\",\\"ts\\":1486970750794,\\"fcnt\\":4368,\\"port\\":1,\\"freq\\":867700000,\\"toa\\":1482,\\"dr\\":\\"SF12 BW125 4/5\\",\\"ack\\":false,\\"gws\\":[{\\"rssi\\":-120,\\"snr\\":-6.5,\\"ts\\":1486970750794,\\"gweui\\":\\"024B0BFFF03054B\\"},{\\"rssi\\":-121,\\"snr\\":0.2,\\"ts\\":1486970750825,\\"gweui\\":\\"B827EBFFF7C1C5F\\"},{\\"rssi\\":-116,\\"snr\\":-3.2,\\"ts\\":1486970750881,\\"rsig\\":[{\\"ant\\":0,\\"chan\\":3,\\"rssic\\":-117,\\"lsnr\\":-3.2},{\\"ant\\":1,\\"chan\\":19,\\"rssic\\":-116,\\"lsnr\\":-3.2}],\\"gweui\\":\\"7076FFFFF0103FA\\"}],\\"data\\":\\"070f2021140a0f5a00\\"},"tech":"L"}
```

Obr. 5.21: Ukázka zprávy formátu JSON [20]

5.4.3 Databáze

Pro ukládání dat bude využito webových služeb společnosti Amazon, tyto služby se označují Amazon Web Services - AWS. Bude využita služba Relational Database Service, ve které poběží instance MySQL databáze. MySQL databáze byla použita z důvodu zkušeností autora práce s touto databází. Pro přenos dat ze zprávy typu JSON odeslané CRA bude využita architektura jenž je zobrazena na obr. 5.22.



Obr. 5.22: Amazon Serverless architektura [21]

Architekturu tvoří 3 části:

- API Gateway
- Lambda funkce

- Databáze

API Gateway

Na API Gateway se můžeme dívat jako na vstupní bránu pro aplikace, které chtějí přistupovat k datům z databáze nebo naopak chtějí do databáze data vkládat. API Gateway tedy představuje endpoint, který je potřeba zadat v CRA IOT portálu, na který se budou posílat zprávy.

Bylo tedy potřeba vytvořit API Gateway podporující protokol REST a metodu POST. Dále bylo třeba napojit metodu POST na lambda funkci a s využitím mapping template předat body json zprávy do lambda funkce.

Lambda funkce

Lambda funkce je řešení, které umožňuje zpracování dat, které přišli přes API Gateway. Výhodou použití lambda funkce je, že není potřeba vytvářet a platit za běžící server.

V případě této práce byla vytvořena lambda funkce v jazyce python, která rozparsuje json řetězec odeslaný Českými radiokomunikacemi a uloží data do databáze.

Zdrojový kód vytvořené lambda funkce je přiložen v příloze.

MySQL databáze

V databázi MySQL byla vytvořena tabulka, do které ukládá lambda funkce data. Tato tabulka obsahuje následující informace:

- ID
- DEVUI
- Teplota
- Vlhkost
- Tlak
- Časová značka

5.5 Tvorba webové aplikace

V této části práce je popsána tvorba webové aplikace, která je určena pro vizualizaci naměřených dat, které jsou uloženy v databázi MySQL. Prvním úkolem bylo definovat požadavky na aplikaci a určit jaké prostředky budou využity k programování aplikace. Následovala volba architektury aplikace, tedy volba návrhového vzoru. Podobně jako u tvorby aplikace pro PC byla důvodem následování návrhového vzoru

snadná úprava oddělených částí aplikace, snazší možnost rozšíření aplikace o další funkce a v neposlední řadě snazší orientace v kódu.

5.5.1 Požadavky na aplikaci

Požadavky na aplikaci jsou shrnuty v následujícím seznamu:

- Připojení k databázi MySQL
- Správa uživatelů, rolí - minimálně Admin a přihlášený uživatel
- Nového uživatele může přidat pouze Admin
- Přihlášený uživatel si může přidávat/mazat zařízení, které identifikuje parametr DevEui
- Zobrazení naměřených hodnot teploty, vlhkosti a tlaku z jednotlivých zařízení v grafu

5.5.2 ASP.NET Core

Aplikace je naprogramována v ASP.NET Core. Hlavním důvodem je autorova zkušenost s .NET, viz. výše popsaná aplikace pro PC. Mezi další důvod je možno zařadit například multiplatformnost ASP.NET Core, tzn. že aplikace napsané v ASP.NET Core mohou fungovat na Windows, Linux i Mac.

5.5.3 Návrhový vzor MVC

Základním kamenem návrhového vzoru MVC je oddělení datového modelu, uživatelského rozhraní a řídicí logiky do 3 nezávislých komponent tak, aby při potřebě změny jedné z těchto komponent pouze minimálně ovlivnilo komponenty ostatní.[22] Tyto komponenty jsou nazývány:

- Models
- Views
- Controllers

Název návrhového vzoru je tedy odvozen z prvních písmen základních 3 komponent, na kterou dělí aplikaci.

Models

Model obsahuje logiku, výpočty, databázové dotazy, validace dat, apod. Model správně neví nic o formátování dat na výstupu. Například při použití objektově relačního mapování, odpovídá model přímo databázové tabulce. Bez použití objektově relačního mapování je potřeba využít tzv. manažera obsahujícího metody k výběru potřebných dat z databáze.[22]

Views

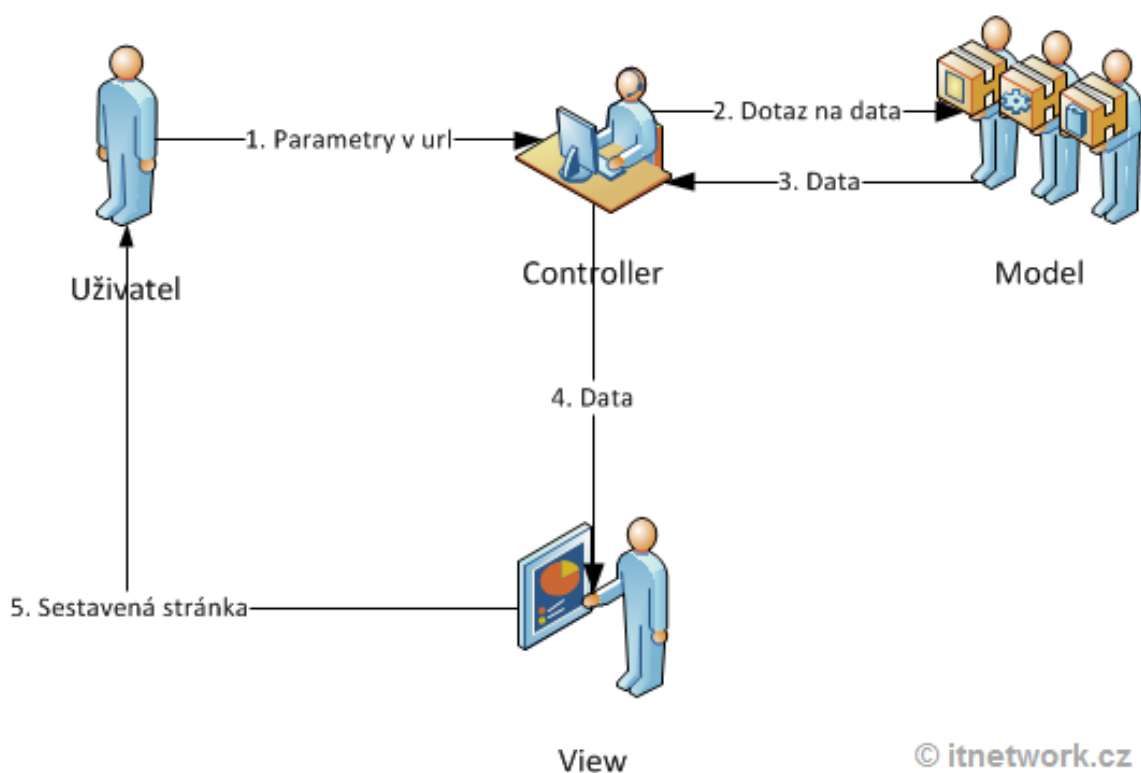
Views mají na starosti formátování výstupu pro uživatele. Často se jedná o šablonu obsahující html stránku s tagy některého ze značkovacích jazyků. View samozřejmě musí obsahovat alespoň minimální množství logiky, např. cykly pro vypsání komentářů. Celkově ale View neví nic o zdroji dat, stará se pouze o zobrazení.[22]

Controllers

Controller reaguje na události a zajišťuje změny v modelu. Na základě změn je následně provedena aktualizace pohledu.[22]

Životní cyklus stránky

Životní cyklus stránky lze popsat diagramem 5.23.



Obr. 5.23: Diagram MVC [22]

Vysvětlení obrázku 5.23 na příkladu[22]:

1. Uživatel zadá do prohlížeče adresu webu a parametry
2. Požadavek zachytí router a podle parametrů rozpozná, který controller je volán
3. Controller podle parametrů rozpozná, co má udělat - např. zobrazit detail uživatele s id 15

4. Controller zavolá model, který v databázi vyhledá uživatele a vrátí jeho data
5. Nakonec je vyrenderováno View. View jsou předány proměnné s příslušnými daty
6. View přijme data od Controlleru a vloží je do připravené šablony

5.5.4 Model

Model je část představující práci s daty, konkrétně s tabulkami databáze. Defaultně je ASP.NET Core určen pro práci s databází MsSQL, nicméně po instalaci odpovídajícího connectoru lze pracovat s libovolnou databází. V tomto případě lze po instalaci MySQL connectoru pracovat s databází MySQL.

db.cs

Třídě db.cs náleží metoda GetDbData, která je určena pro čtení naměřených dat náležících konkrétnímu zařízení. Diagram třídy db.cs je na obr. 5.24.

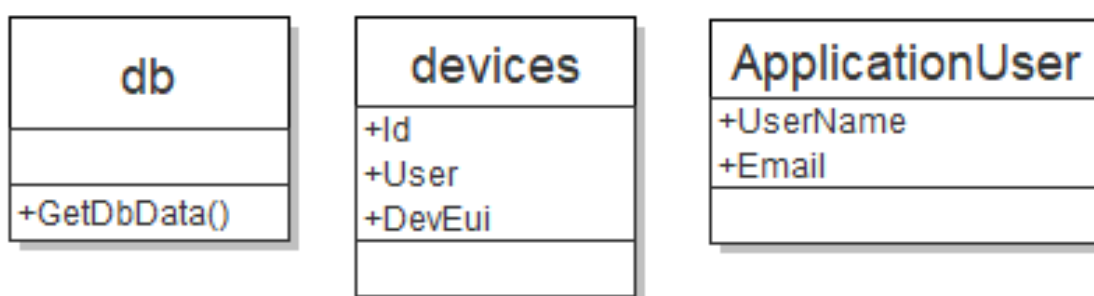
devices.cs

Třída devices.cs sdružuje zařízení patřící konkrétnímu uživateli. Tabulka má tedy 3 sloupce:

- Id
- User
- DevEui

ApplicationUser.cs

Třída ApplicationUser dědí ze třídy IdentityUser a obsahuje 2 proměnné, UserName a Email.

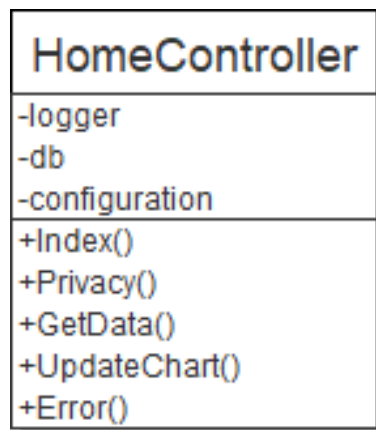


Obr. 5.24: Diagramy tříd modelů

5.5.5 Controllers

HomeController.cs

Třídě HomeController náleží 2 akce. První akcí je Index, jejímž výsledkem je zobrazení hlavní stránky aplikace, tzv. Home. Druhou akcí je Privacy, jejímž výsledkem je zobrazení grafu s naměřenými hodnotami příslušné veličiny. Třída HomeController dále obsahuje 2 metody, GetData a UpdateChart, pro práci s modelem db.cs, tedy s naměřenými daty. Tato data v podobě JSON poskytuje příslušnému View.



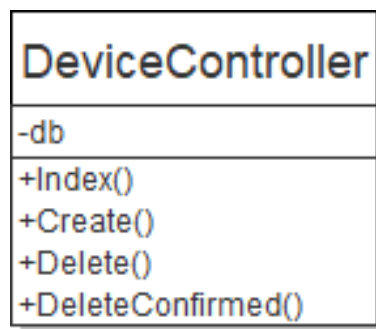
Obr. 5.25: Diagramy třídy HomeController

DeviceController.cs

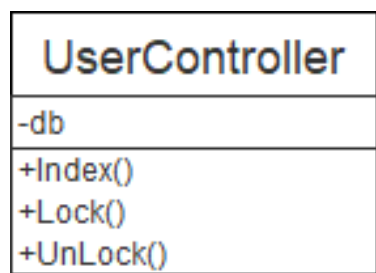
Třída DeviceController obsahuje akci Index, jejímž výsledkem je zobrazení zařízení, které náleží přihlášenému uživateli. Třída DeviceController dále obsahuje GET a POST akci Create, jejímž výsledkem je vytvoření nového zařízení pro přihlášeného uživatele. Dále třída obsahuje GET a POST akci Delete, pomocí které přihlášený uživatel může smazat své dříve vytvořené zařízení. Třída DeviceController je přístupná pouze přihlášenému uživateli s rolí Customer.

UserController.cs

Třída UserController obsahuje akci Index, jejímž výsledkem je zobrazení registrovaných uživatelů. Dále třída DeviceController obsahuje akce Lock a Unlock, které slouží k zamčení a odemčení účtu příslušného uživatele. Pokud je účet příslušného uživatele zamčený, uživatel se nebude moci přihlásit do aplikace.



Obr. 5.26: Diagramy třídy DeviceController



Obr. 5.27: Diagramy třídy UserController

5.5.6 Views

Views je potřeba rozdělit dle výše zmíněných použitých uživatelských rolí a Controlleru přidruženého k View.

Home - Index

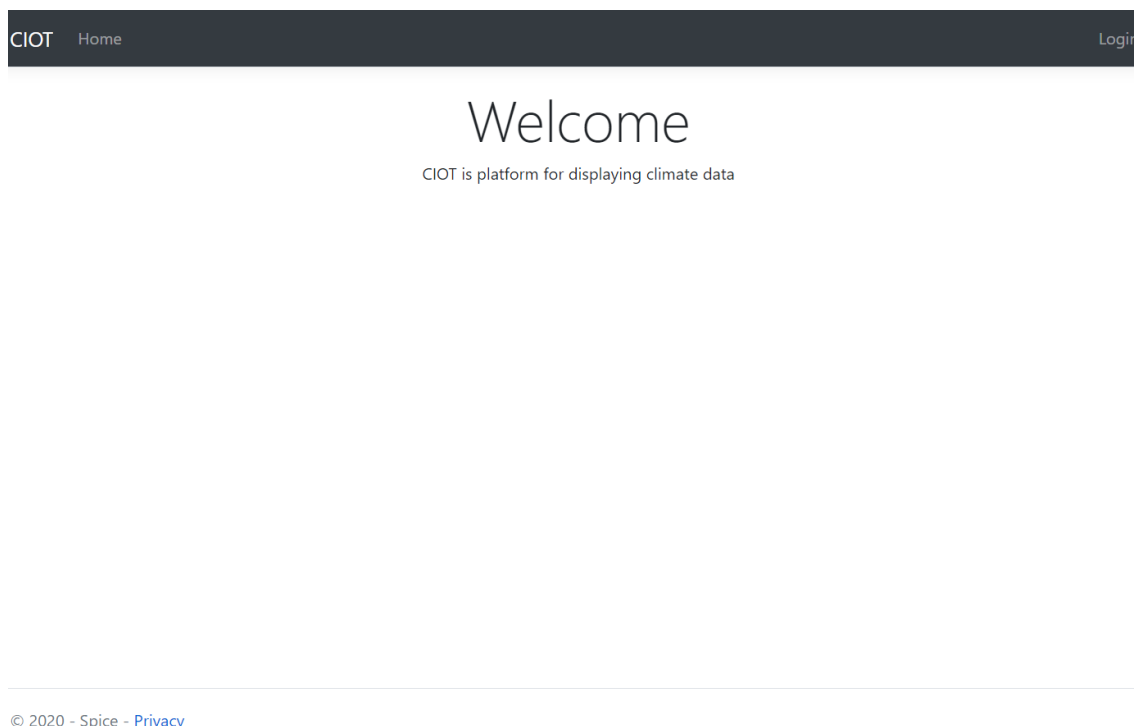
Ukázka hlavní stránky aplikace, kterou může vidět i nepřihlášený uživatel je na obr. 5.28.

Device - Index

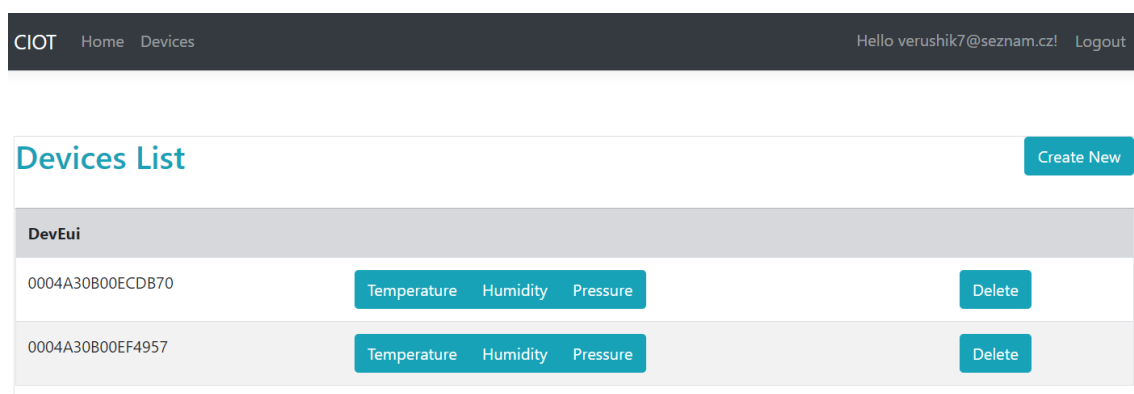
Pokud se uživatel přihlásí s rolí Customer, má navíc k dispozici na hlavní navigační liště položku devices. Pokud uživatel na tuto položku klikne, přejde aplikace na odpovídající View a zobrazí seznam zařízení, které uživateli náleží, viz. 5.29.

Pokud uživatel klikne na tlačítko Create New, otevře se mu nové View - Create, viz. obr. 5.30. Zde uživatel může vytvořit nové zařízení podle unikátního klíče DevEui.

Pokud naopak uživatel klikne na tlačítko Delete, otevře se nové View - Delete, viz. obr. 5.31, kde uživatel může potvrdit smazání zařízení ze svého seznamu. Dále jsou na stránce se seznamem zařízení 3 tlačítka odpovídající měřeným veličinám, tzn. teplotě, vlhkosti a tlaku. Po kliknutí na jedno z těchto veličin se otevře View,



Obr. 5.28: Ukázka hlavní stránky aplikace



Obr. 5.29: Ukázka seznamu zařízení náležících uživateli

na kterém je zobrazen graf s průběhem příslušné veličiny v čase, viz. obr. 5.32. Graf má několik možností zoomování. Mezi tyto možnosti patří volba celého průběhu v čase, který je v databázi nebo 1 minuta, 1 hodina, 1 den, 1 měsíc, 1 rok. Dále je možno zoomovat na libovolný časový úsek pomocí posuvníků pod grafem. Dále je možno zobrazit graf na celou stránku pro lepší rozlišitelnost.

CIOT

Home

Devices

Hello verushik7@seznam.cz!

Logout

Create device

DevEui

Create

Back to List

Obr. 5.30: Ukázka View pro vytvoření nového zařízení

CIOT

Home

Devices

Hello verushik7@seznam.cz!

Logout

Delete device

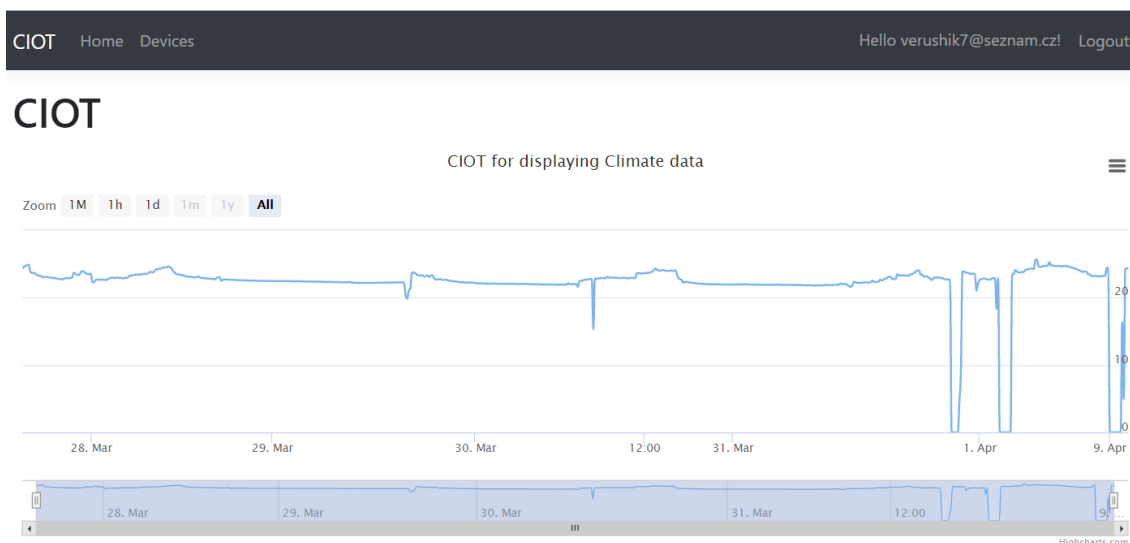
DevEui

0004A30B00EF4957

Delete

Back to List

Obr. 5.31: Ukázka View pro smazání zařízení

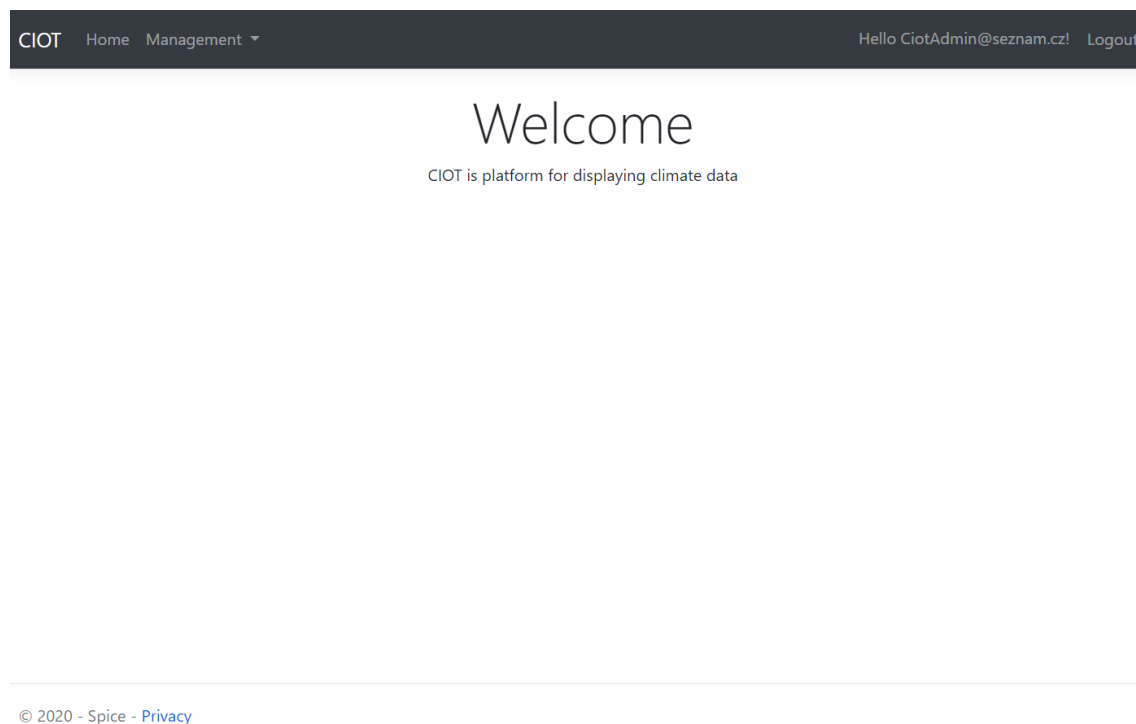


Obr. 5.32: Ukázka grafu teploty

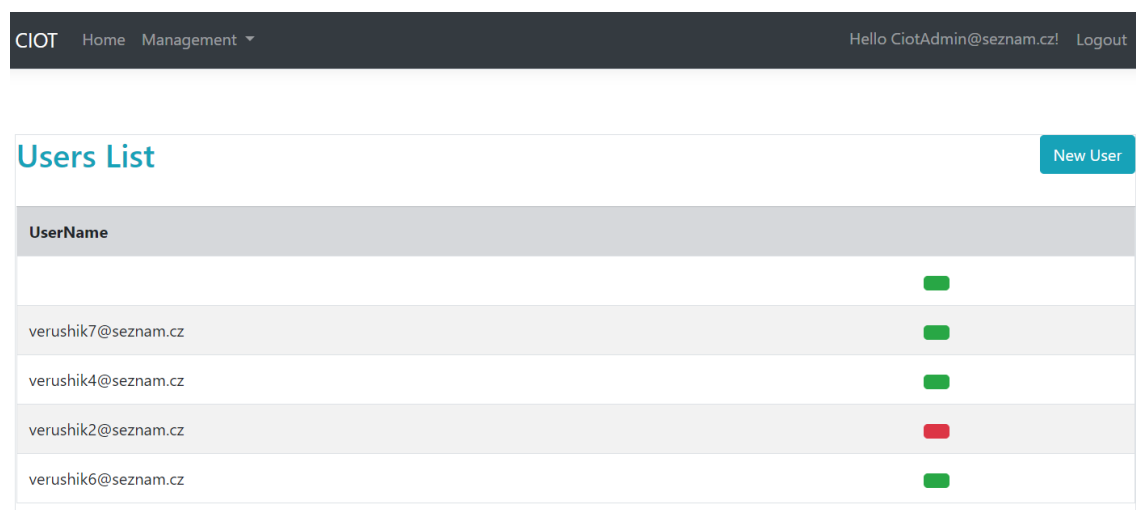
Admin - Index

Pokud se přihlásí uživatel s rolí Admin, má na hlavní navigační liště místo položky devices položku Management v podobě vysouvacího menu, viz. obr. 5.33, kde je možno kliknout na položku Users. Po kliknutí na položku Users se zobrazí View, kde je zobrazen seznam zaregistrovaných uživatelů, viz. obr. 5.34. U jednotlivých

uživatelů je zobrazen na řádku zelený nebo červený obdélník. Zelený obdélník symbolizuje odemknutý účet, naopak červený obdélník symbolizuje zamknutý účet. Stav účtu lze změnit kliknutím na obdélník. Dále je možno kliknout na tlačítko Create New, což vyvolá zobrazení nového View, viz. obr. 5.35, kde lze zaregistrovat nového uživatele.



Obr. 5.33: Ukázka domovské obrazovky uživatele s rolí Admin



Obr. 5.34: Ukázka View se seznamem zaregistrovaných uživatelů

CIOT

Home

Management ▾

Hello CiotAdmin@seznam.cz!Logout

Register

Create a new account.

Email

Password

Confirm password

☒ Customer

☐ Manager

Register

Obr. 5.35: Ukázka registrace nového uživatele

5.5.7 Publikování aplikace

Aby byla aplikace přístupná okolnímu světu z webového prohlížeče, je potřeba ji publikovat. Za tímto účelem byl zřízen účet v Microsoft Azure. Následně byla aplikace publikována s využitím nástroje integrovaného v Microsoft Visual Studiu. Adresa aplikace je: <https://spice20200413044715.azurewebsites.net>

6 Závěr

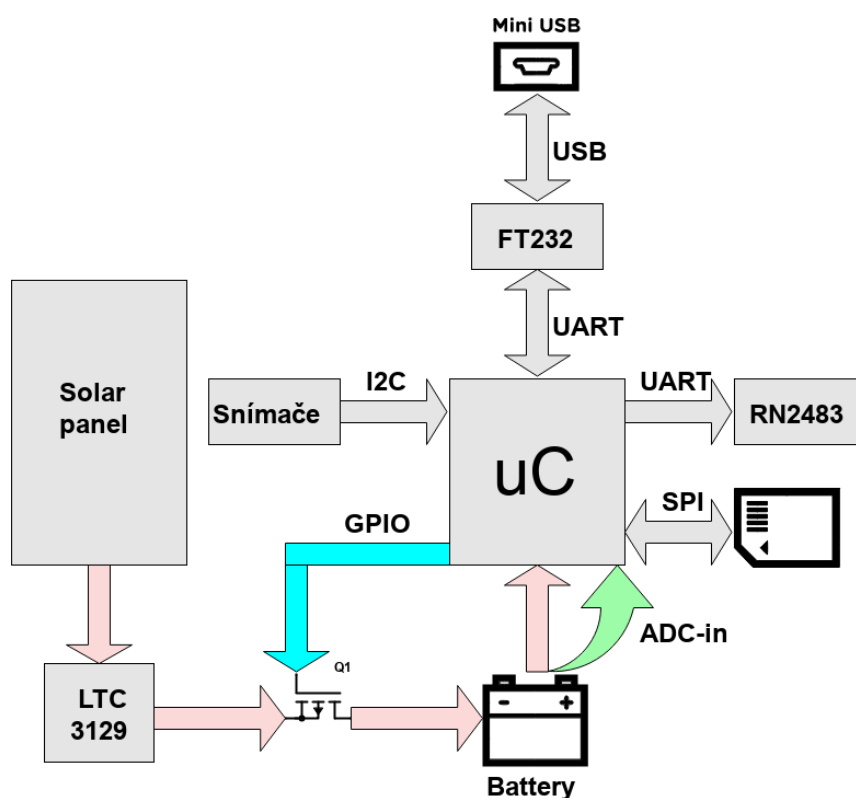
Závěr práce je rozdělen do několika následujících částí.

6.1 Koncept záznamníku

Nejprve byl proveden průzkum trhu a na jeho základě vznikaly požadavky na záznamník, viz. následující výčet.

- Solární napájení
- Mikrokontrolérem řízený systém
- Sběr dat teploty, vlhkosti a tlaku
- Ukládání dat na paměťovou kartu
- Využití některé z IOT technologií
- Konfigurace záznamníku pomocí PC aplikace přes USB
- Vizualizace naměřených dat

Na základě průzkumu trhu, požadavků zadání a výše popsanych požadavků byl vytvořen koncept záznamníku, viz. obr. 6.1. Následně byly vybrány snímače pro



Obr. 6.1: Finální koncept záznamníku i s hlavními obvody

měření klimatických dat i ostatní komponenty potřebné pro realizaci záznamníku. Koncept, i s vybranými hlavními součástkami, je možno shrnout v následujícím výčtu:

- Solární panel je připojen na vstup obvodu LTC-3129
- Výstup obvodu LTC-3129 dobíjí Lithium-titanovou baterii - při určitém napětí baterie, mikrokontrolér přeruší dobíjení
- Napětí z baterie je pomocí step-up měniče MAX1676 zvýšeno na 3.3V.
- Mikrokontrolér STM32L151 komunikuje se snímači přes I2C. Dále komunikuje s paměťovou kartou přes SPI. Dále mikrokontrolér komunikuje s LoRa modulem RN2483 pomocí UART. Modul RN2483 se stará o odesílání dat pomocí LoRa technologie. Pomocí UART mikrokontrolér komunikuje i s obvodem FT232RL, který je určen pro komunikaci s PC přes USB.

Následně byla určena energetická bilance, přičemž průměrný odběr zařízení byl stanoven 0.153 mW.

6.2 HW

Elektronika záznamníku, tzn. schéma a desky plošných spojů, byla navržena pomocí softwaru Eagle 6.6.0. Příslušná schémata a dps jsou k dispozici v příloze.

6.3 SW

Bylo potřeba napsat software pro mikrokontrolér, pro PC aplikaci určenou ke konfiguraci zařízení přes USB. Následně bylo potřeba vyřešit vizualizaci naměřených, přes LoRa odeslaných dat.

6.3.1 SW pro mikrokontrolér

SW pro mikrokontrolér byl napsán v jazyce C s využitím knihoven HAL. Činnost mikrokontroléru bude zkráceně shrnuta v následujících větách. Mikrokontrolér je v režimu spánku. Z režimu spánku může být vzbuzen připojením zařízení k PC. Pro komunikaci s PC byl implementován vlastní komunikační protokol, popsáný v Komunikační protokol pro komunikaci s PC. Z PC může mikrokontrolér dostat příkaz k nastavení času nebo ke spuštění měření. Po spuštění měření je z režimu spánku mikrokontrolér buzen periodicky od RTC alarmu jednou za min. pro odměření dat pomocí připojených snímačů a jednou za 7min. pro odeslání dat pomocí LoRa technologie a pro uložení dat na paměťovou kartu. Detailní popis SW pro mikrokontrolér je v kapitole Program mikrokontroléru.

6.3.2 PC aplikace

PC aplikace s grafickým uživatelským rozhraním pro konfiguraci záznamníku přes USB byla napsána v jazyce C# s využitím WPF. Aplikace je napsána podle návrhového vzoru MVVM. Detailní popis aplikace je v předchozích kapitolách. Aplikace pracuje s virtuálním COM Portem s parametry komunikace: Baudrate-38400, Parity-None, Stopbits-1. Aplikace periodicky vyčítá hodnoty teploty, vlhkosti, tlaku a stav zařízení, tzn. časovou značku a zda je měření spuštěno. V aplikaci je možno nastavit časovou značku pro zařízení a spustit měření. Pro komunikaci byl navržen vlastní, jednoduchý komunikační protokol, popsáný v kapitole Komunikační protokol pro komunikaci s PC. Podrobný popis aplikace je v kapitole Program pro ovládání zařízení z PC.

6.3.3 Vizualizace naměřených dat

Naměřená data jsou z mikrokontroléru pomocí LoRa modulu RN2483 posílány do cloudu Českých radiokomunikací. Nicméně v tomto cloudu není možno data vizualizovat. Proto bylo využito možnosti automaticky přeposílat přijatá data na vlastní endpoint. Jako vlastní endpoint slouží API Gateway od Amazon Web Services, která pomocí Lambda funkce, napsané v jazyce python, ukládá data do zřízené MySQL databáze. Pro vizualizaci dat uložených v této databázi byla napsána webová aplikace.

6.3.4 Webová aplikace

Webová aplikace byla napsána v ASP.NET Core s využitím návrhového vzoru MVC. V aplikaci existují 2 role: admin, customer. Pokud se přihlásí uživatel s rolí admin, má k dispozici správu účtů, tzn. že může přidávat nové uživatele s rolí admin nebo customer, a nebo některému z uživatelů zamknout, resp. odemknout účet. Pokud se přihlásí uživatel s rolí customer, má k dispozici svá zařízení. Svá zařízení může uživatel přidávat nebo mazat. U každého svého zařízení si může uživatel zobrazit naměřená data. Naměřená data se zobrazují v grafu, přičemž je možno volit zobrazený časový interval. Detailní popis aplikace je v kapitole Tvorba webové aplikace.

6.4 Zhodnocení

Na základě průzkumu trhu a zadání byl navržen koncept záznamníku klimatických dat se solárním napájením. Následně byla zpracována energetická bilance. Poté byla

navržena elektronika záznamníku. Při první realizaci dps došlo k nevratnému poškození dps při pokusu o odpájení mikrokontroléru. Nicméně s využitím mikrokontroléru napájeného na univerzální dps byly oživeny ostatní obvody na dps a díky tomu byly v další verzi dps odstraněny nedostatky, které vznikly při návrhu. Následně byl napsán program pro mikrokontrolér, program v C# pro PC, který konfiguruje zařízení přes USB a byla vyřešena vizualizace naměřených dat pomocí služeb Amazon Web Services a napsané webové databázové aplikace v ASP.NET Core. Tímto byly splněny všechny body zadání.

6.4.1 Návrhy do budoucna

Z HW hlediska by bylo dobré zjistit, proč step-up měnič MAX1676 pracuje s mnohem nižší efektivitou než bylo očekáváno a než udává příslušný katalogový list součástky, popř. nahradit tento obvod jiným step-up měničem. Dále by bylo dobré navrhnout krabičku pro zařízení na potřebné rozměry a vytisknout na 3D tiskárně, neboť na trhu krabička s ideálními rozměry nebyla nalezena a musela být použita krabička větší. Ze SW hlediska se nabízí přidat uživateli možnost volby periody měření, tzn. že by uživatel před spuštěním měření v PC aplikaci nastavil periodu měření. Nyní je perioda měření 1min. bez možnosti volby.

Literatura

- [1] Lascar Electronics: *EL-USB-2-LCD* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<<https://www.lascarelectronics.com/media/1505/easylog-data-logger-el-usb-2-lcd.pdf>>.
- [2] Gemini dataloggers: *Tinytag Plus 2 - TGP-4020* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<<http://gemini2.assets.d3r.com/pdfs/original/3747-tgp-4020.pdf>>.
- [3] Rotronic Instruments: *HL-NT2-DP* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<<https://docs.rs-online.com/c7d5/0900766b80f29d83.pdf>>.
- [4] Sensirion: *SHT85* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<<https://cdn.sos.sk/productdata/8d/94/c6c9b77b/sht-85.pdf>>.
- [5] Bosch: *BMP280* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<<https://cz.mouser.com/datasheet/2/783/BST-BMP280-DS001-1509562.pdf>>.
- [6] ANALOG DEVICES: *Interfacing an SD Card Through the ADuCM3027/ADuCM3029 SPI Bus* [online]. [cit. 2. 12. 2019]. Dostupné z URL:
<<https://www.analog.com/media/en/technical-documentation/application-notes/AN-1443.pdf>>.
- [7] Z. Lu, J. Li and Y. Zhang, "The Reading/Writing SD Card System Based on FPGA,; 2010 First International Conference on Pervasive Computing, Signal Processing and Applications, Harbin, 2010, pp. 419-422
- [8] FTDI chip: *TTL-232R-PCB* [online katalogový list]. [cit. 2. 12. 2019]. Dostupné z URL:
<https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_PCB.pdf>.
- [9] Orange Connected Objects & Partnerships: *LoRa Device Developer Guide* [online]. [cit. 29. 10. 2019]. Dostupné z URL:
<<https://developer.orange.com/wp-content/uploads/LoRa-Device-Developer-Guide-Orange.pdf>>.

- [10] Microchip Technology: *RN2483* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<http://www.farnell.com/datasheets/2648020.pdf?_ga=2.264671883.529461036.1569437728-1929573064.1562862675&_gac=1.207826598.1569508947.EAIAIQobChMIss3Zttzu5AIVVODtCh3KjAk0EAAYASAAEgLQRPD_BwE>.
- [11] Microchip Technology: *RN2483 LoRa Technology Module Command Reference User's Guide* [online katalogový list]. [cit. 23. 10. 2019]. Dostupné z URL:
<<https://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf>>.
- [12] sigfox: *Sigfox Technical Overview* [online]. [cit. 29. 10. 2019]. Dostupné z URL:
<<https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>>.
- [13] X. Wu, Z. Mei, C. Hu, C. Zhu and J. Sun, "Temperature Performance Comparative Analysis of Different Power Batteries," 2016 IEEE Vehicle Power and Propulsion Conference (VPPC), Hangzhou, 2016, pp. 1-6.
- [14] STMicroelectronics: *RM0038* [Reference manual].
- [15] maxim integrated: *MAX1674/MAX1675/MAX1676* [online katalogový list]. [cit. 23. 4. 2020]. Dostupné z URL:
<<https://datasheets.maximintegrated.com/en/ds/MAX1674-MAX1676.pdf>>.
- [16] elm-chan.org: *FatFs - Generic FAT Filesystem Module* [online]. [cit. 27. 4. 2020]. Dostupné z URL:
<http://elm-chan.org/fsw/ff/00index_e.html>.
- [17] stm32f4-discovery.net: *Library 21- Read SD card with FatFs on STM32F4* [online]. [cit. 27. 4. 2020]. Dostupné z URL:
<<http://stm32f4-discovery.net/2014/07/library-21-read-sd-card-fatfs-stm32f4xx-devices/>>.
- [18] Dps-az.cz: *Techniky programování embedded systémů* [online]. [cit. 23. 11. 2019]. Dostupné z URL:
<<https://www.dps-az.cz/vyvoj/id:6511/techniky-programovani-embedded-systemu>>.

- [19] dotnetportal.cz: *mvvm: model-view-viewmodel* [online]. [cit. 14. 04. 2020]. Dostupné z URL:
<<https://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>>.
- [20] České Radiokomunikace a.s.: *Dokumentace rozhraní pro zasílání IoT zpráv* [Manual].
- [21] dev.to: *Deploying a Serverless Image Service on AWS with CloudFront, Api Gateway, S3, AWS Lambda, and Thumbor* [online]. [cit. 23. 4. 2020]. Dostupné z URL:
<<https://dev.to/astuyve/deploying-a-serverless-image-service-on-aws-with-cloudfront-api-gateway-s3-aws-lambda-and-thumbor-378l>>.
- [22] itnetwork.cz: *MVC architektura* [online]. [cit. 20. 04. 2020]. Dostupné z URL:
<<https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>>.

Seznam příloh

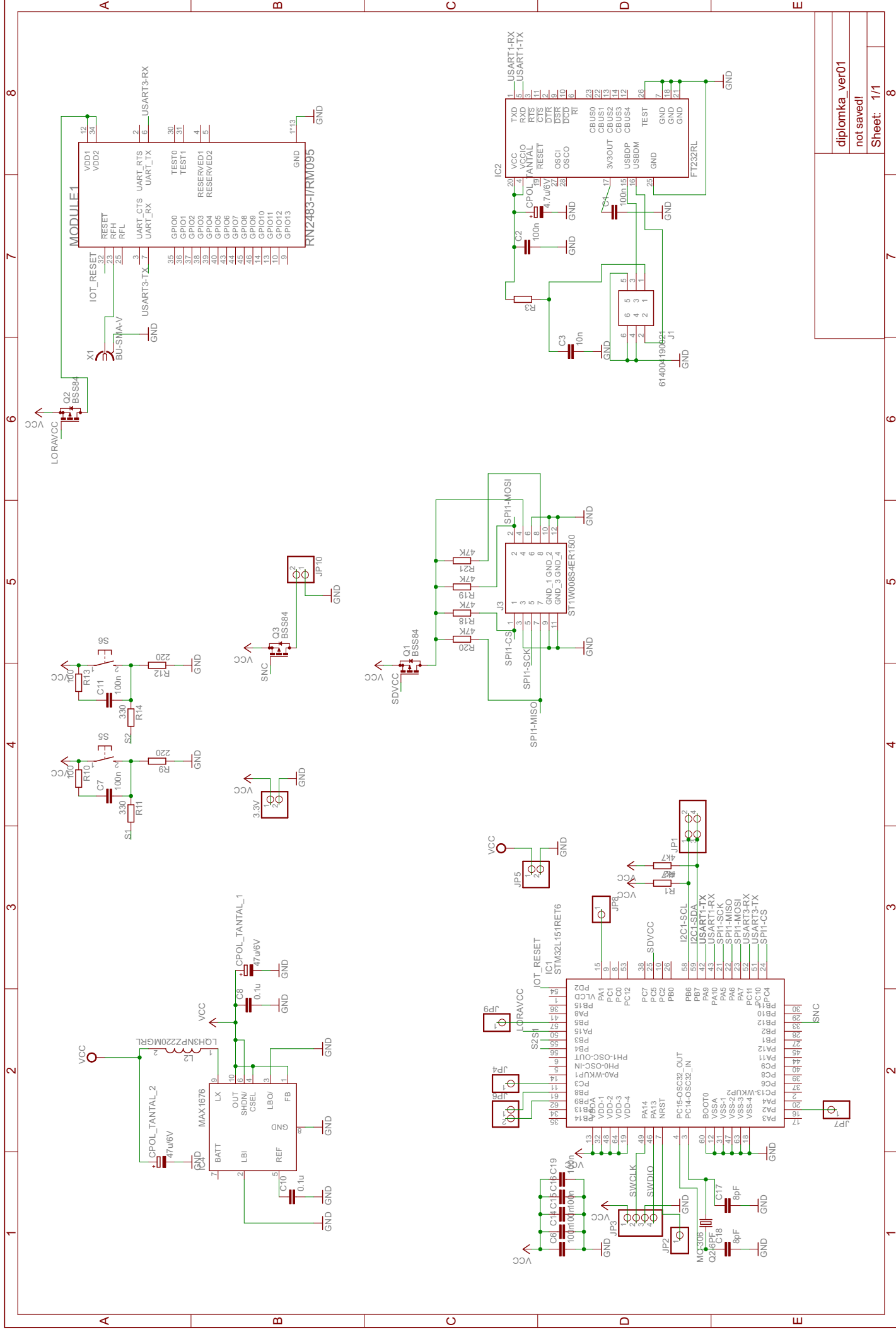
A	Schéma zařízení	91
A.1	Schéma hlavní desky	91
A.2	Schéma LTC-3105	91
A.3	Schéma LTC-3129	91
B	Desky plošných spojů	95
B.1	DPS hlavní desky	95
B.1.1	Horní strana	95
B.1.2	Spodní strana	95
B.1.3	Osazovací schéma	95
B.2	DPS LTC-3105	99
B.2.1	DPS	99
B.2.2	Osazovací schéma	99
B.3	DPS LTC-3129	102
B.3.1	DPS	102
B.3.2	Osazovací schéma	102
C	Seznam součástek	105
C.1	Hlavní deska	105
C.2	LTC-3105	106
C.3	LTC-3129	106
D	Obsah přiloženého CD	107

A Schéma zařízení

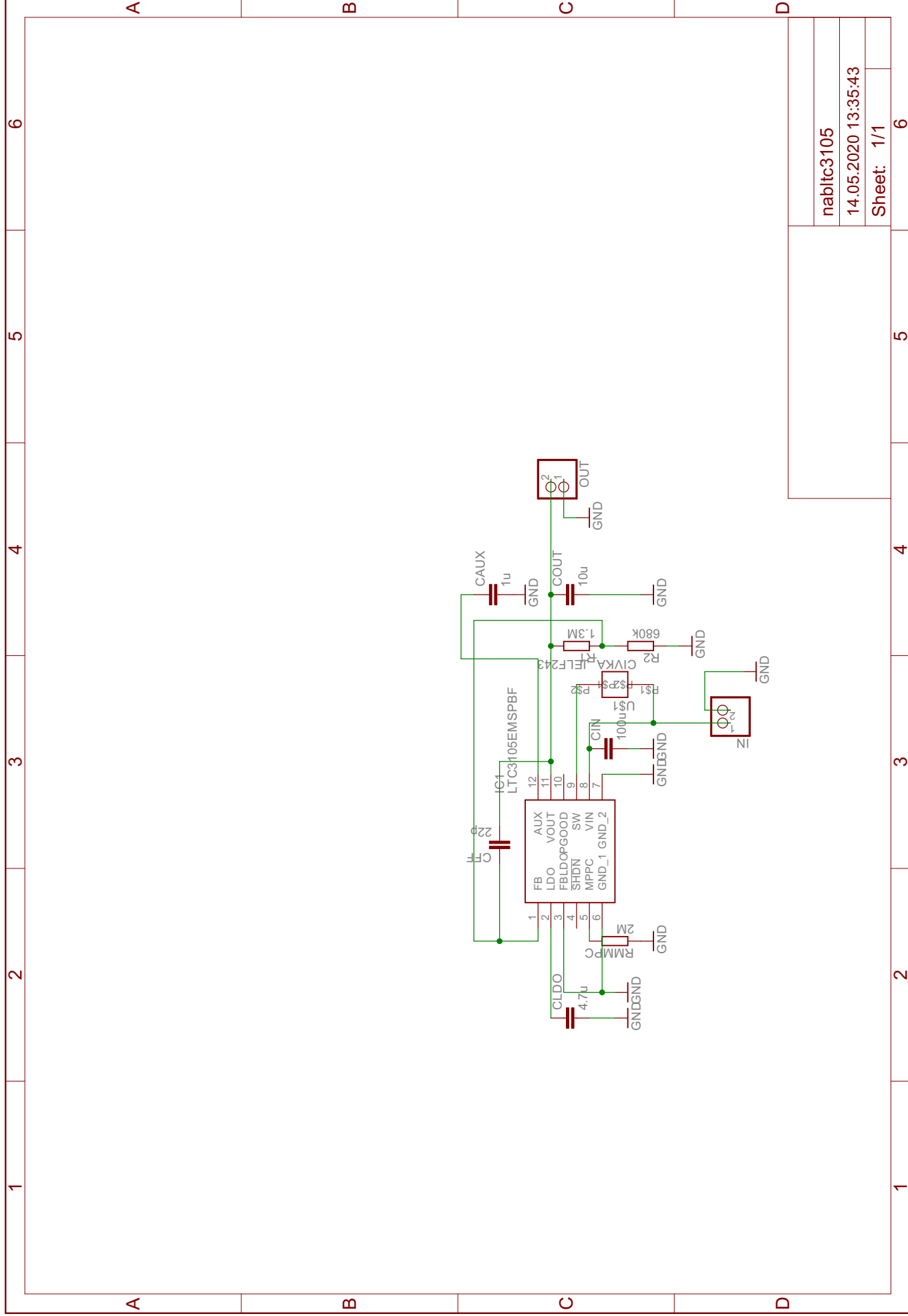
A.1 Schéma hlavní desky

A.2 Schéma LTC-3105

A.3 Schéma LTC-3129



diplomka_ver01	8
not saved!	7
Sheet: 1/1	6
	5
	4
	3
	2
	1



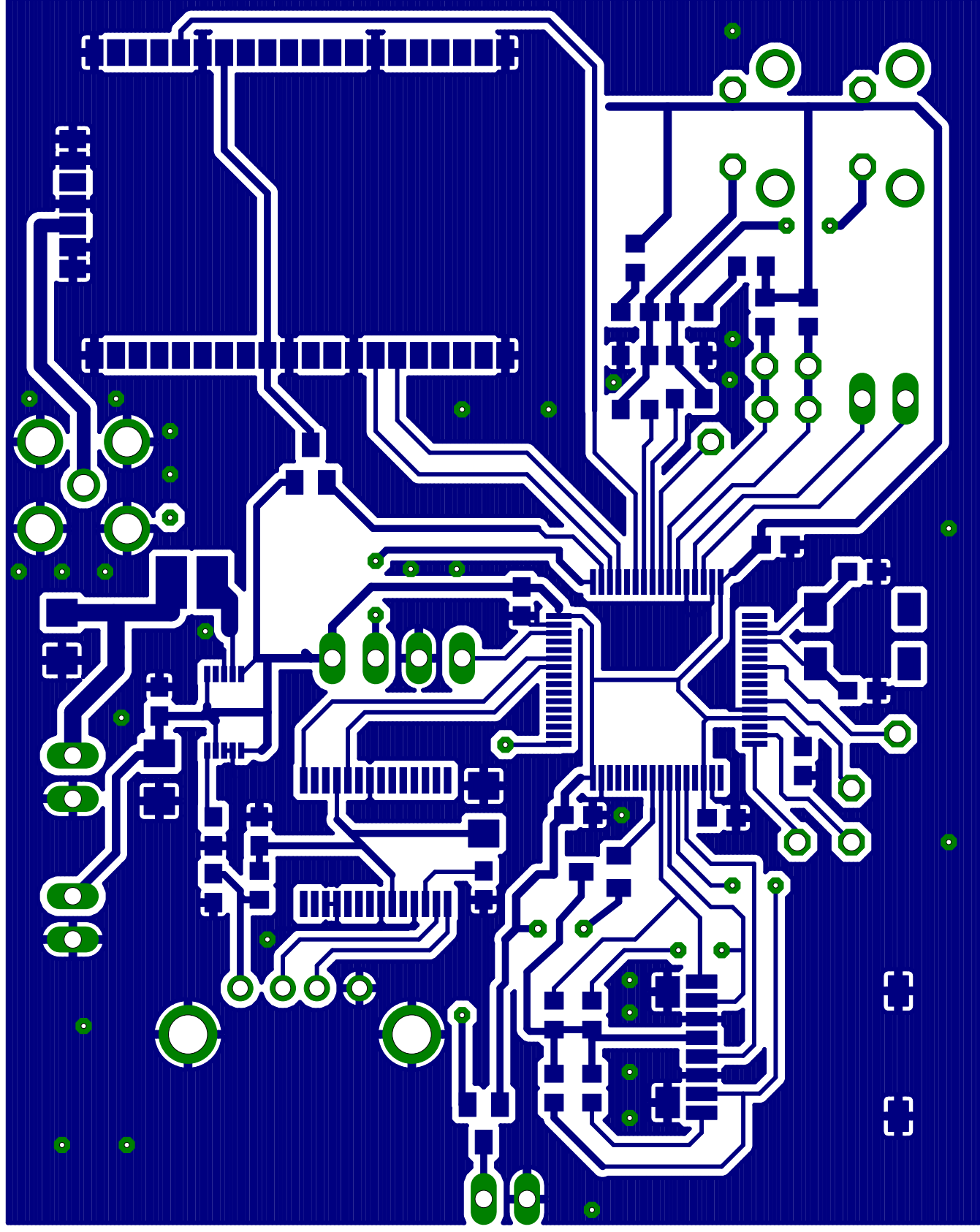
B Desky plošných spojů

B.1 DPS hlavní desky

B.1.1 Horní strana

B.1.2 Spodní strana

B.1.3 Osazovací schéma

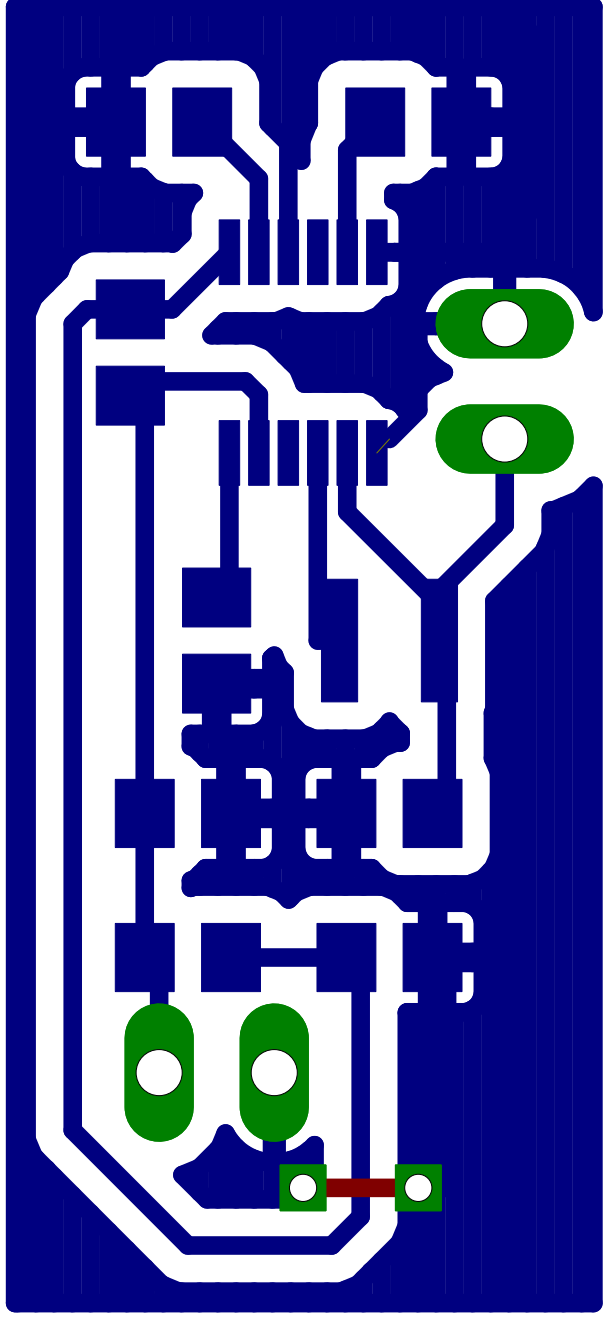


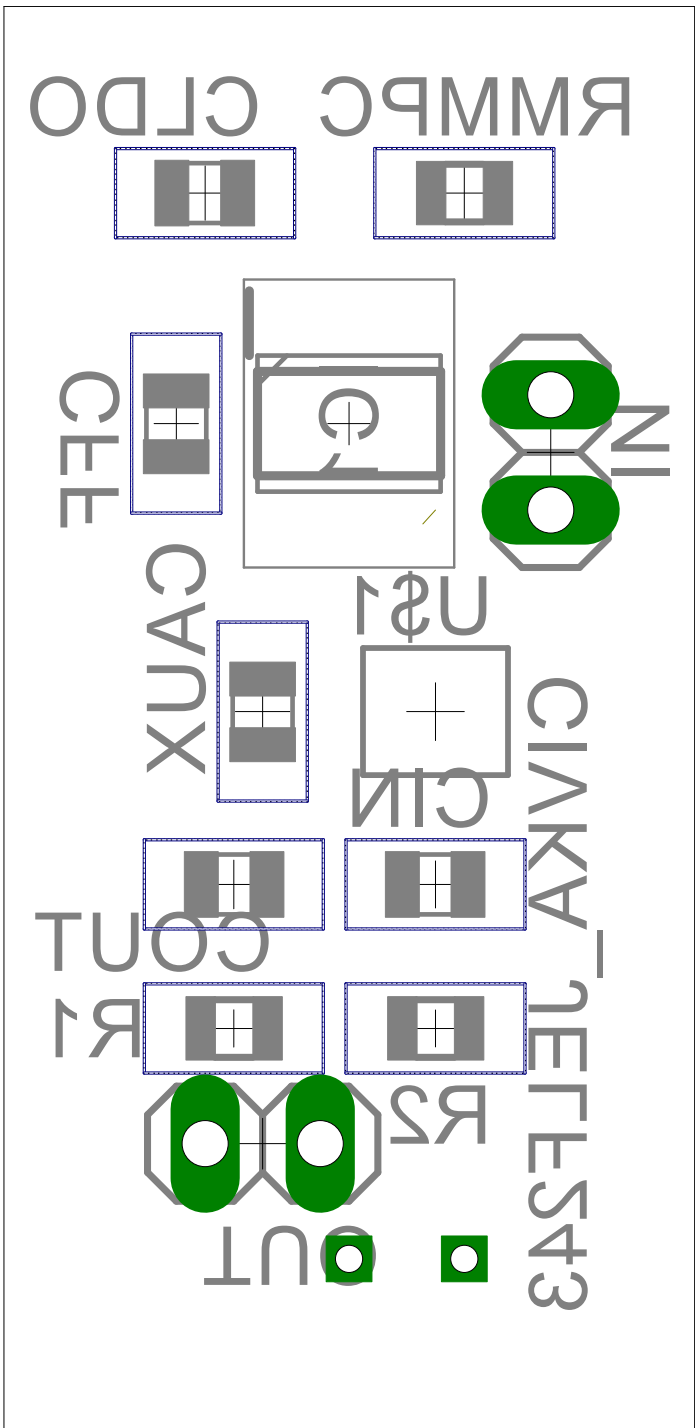


B.2 DPS LTC-3105

B.2.1 DPS

B.2.2 Osazovací schéma

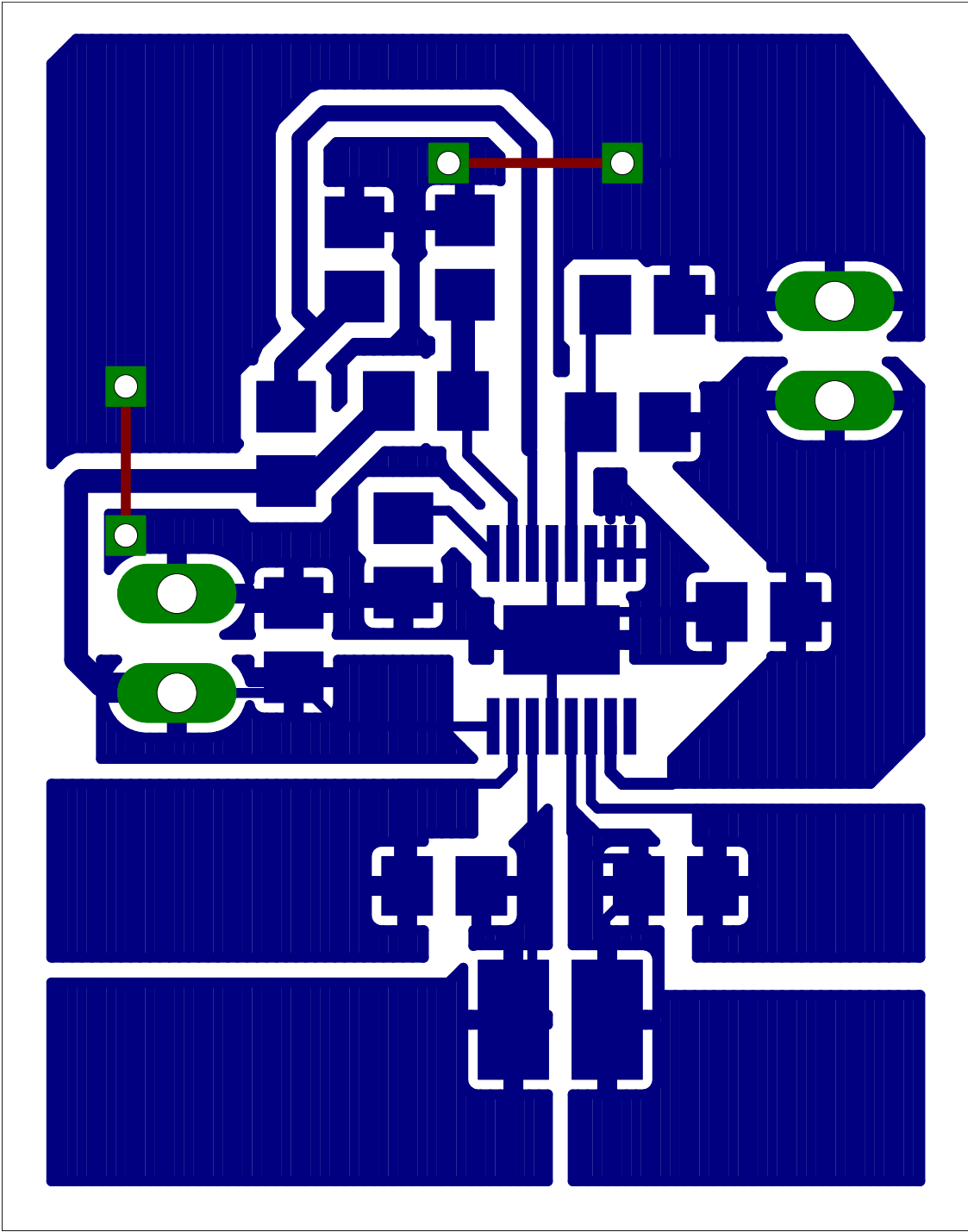


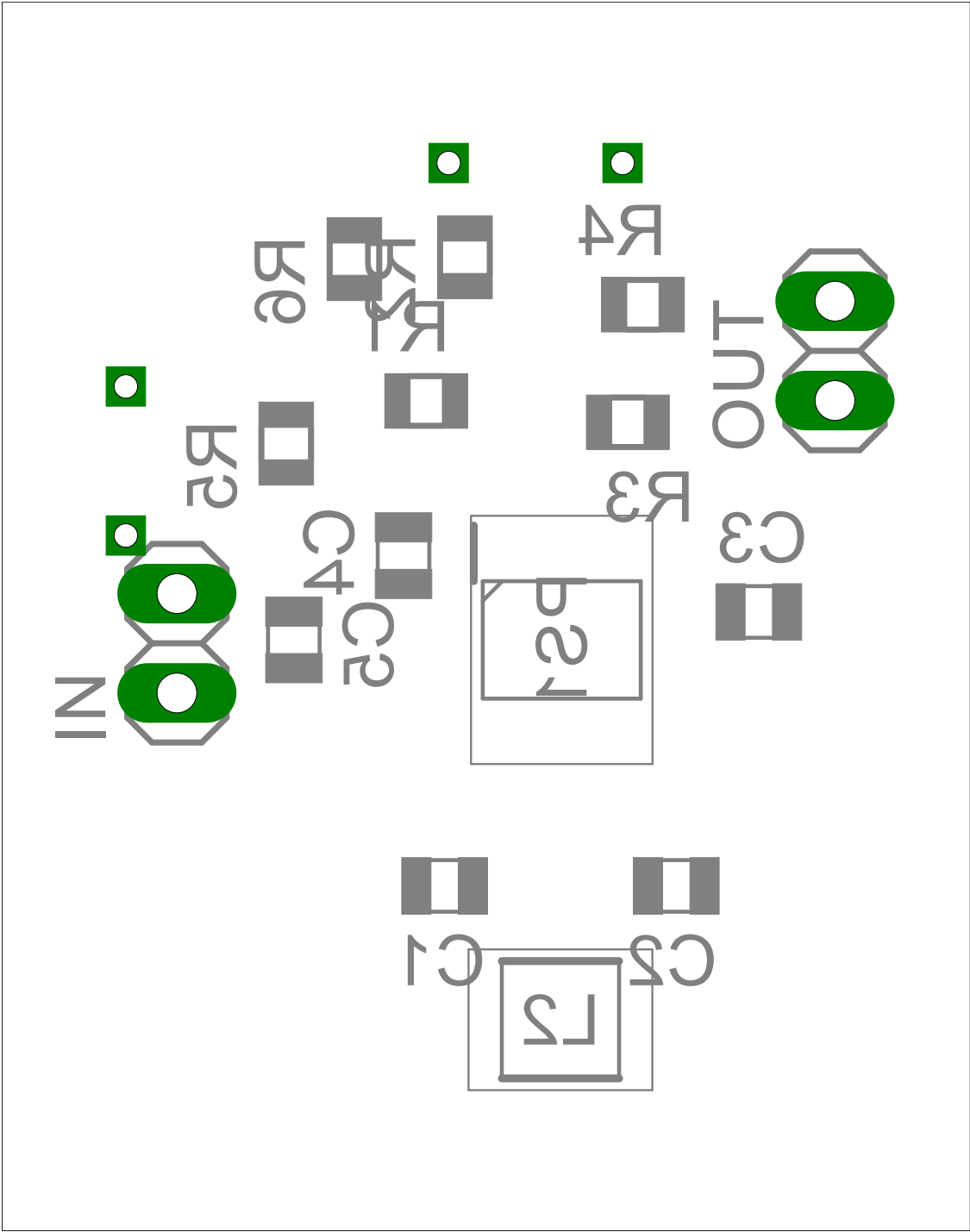


B.3 DPS LTC-3129

B.3.1 DPS

B.3.2 Osazovací schéma





C Seznam součástek

C.1 Hlavní deska

Part	Value	Device	Package	Library
3.3V		PINHD-1X2	1X02	pinhead
C1	100n	C-EUC0603	C0603	rcl
C2	100n	C-EUC0603	C0603	rcl
C3	10n	C-EUC0603	C0603	rcl
C6	100n	C-EUC0603	C0603	rcl
C7	100n	C-EUC0603	C0603	rcl
C8	0.1u	C-EUC0603	C0603	rcl
C10	0.1u	C-EUC0603	C0603	rcl
C11	100n	C-EUC0603	C0603	rcl
C14	100n	C-EUC0603	C0603	rcl
C15	100n	C-EUC0603	C0603	rcl
C16		C-EUC0603	C0603	rcl
C17	8pF	C-EUC0603	C0603	rcl
C18	8pF	C-EUC0603	C0603	rcl
C19	100n	C-EUC0603	C0603	rcl
CPOL-TANTAL	4.7u/6V	C-EUC1206	C1206	rcl
CPOL-TANTAL-1	47u/6V	C-EUC1206	C1206	rcl
CPOL-TANTAL-2	47u/6V	C-EUC1206	C1206	rcl
IC1	STM32L151RET6	STM32L151RET6	QFP50P1200X1200X160-64N	SamacSys-Parts
IC2	FT232RL	FT232RL	SSOP28	ftdichip
IC4	MAX1676	MAX1676	UMAX10	maxim
J1	614004190021	614004190021	USB-2.0-TYPE-A	SamacSys-Parts
J3	ST1W008S4ER1500	ST1W008S4ER1500	ST1W008S4ER1500	SamacSys-Parts
JP1		PINHD-2X2	2X02	pinhead
JP2		PINHD-1X1	1X01	pinhead
JP3		PINHD-1X4	1X04	pinhead
JP4		PINHD-1X1	1X01	pinhead
JP5		PINHD-1X2	1X02	pinhead
JP6		PINHD-1X2	1X02	pinhead
JP7		PINHD-1X1	1X01	pinhead
JP8		PINHD-1X1	1X01	pinhead
JP9		PINHD-1X1	1X01	pinhead
JP10		PINHD-1X2	1X02	pinhead
L2	LQH3NPZ220MGRL	LQH3NPZ220MGRL	INDPM3030X100N	SamacSys-Parts
MODULE1	RN2483-1/RM095	RN2483-1/RM095	MICROCHIP-RN2483-1/RM095	RN2483-1-RM095
Q1	BSS84	BSS84	SOT23	transistor-small-signal
Q2	BSS84	BSS84	SOT23	transistor-small-signal
Q2-6PF	MC-306	MC-306	MC-306	crystal
Q3	BSS84	BSS84	SOT23	transistor-small-signal
R1	4k7	R-EU-R0603	R0603	rcl
R2	4k7	R-EU-R0603	R0603	rcl
R3		R-EU-R0603	R0603	rcl
R9	220	R-EU-R0603	R0603	rcl
R10	100	R-EU-R0603	R0603	rcl
R11	330	R-EU-R0603	R0603	rcl
R12	220	R-EU-R0603	R0603	rcl
R13	100	R-EU-R0603	R0603	rcl
R14	330	R-EU-R0603	R0603	rcl
R18	47K	R-EU-R0603	R0603	rcl
R19	47K	R-EU-R0603	R0603	rcl
R20	47K	R-EU-R0603	R0603	rcl
R21	47K	R-EU-R0603	R0603	rcl
S5		31-XX	B3F-31XX	switch-omron
S6		31-XX	B3F-31XX	switch-omron
X1	BU-SMA-V	BU-SMA-V	BU-SMA-V	con-coax

C.2 LTC-3105

Part	Value	Device	Package	Library
CAUX	1u	C-EUC0805	C0805	rcl
CFE	22p	C-EUC0805	C0805	rcl
CIN	100u	C-EUC0805	C0805	rcl
CLDO	4.7u	C-EUC0805	C0805	rcl
COUT	10u	C-EUC0805	C0805	rcl
IC1	LTC3105EMSPBF	LTC3105EMSPBF	SOP65P490X110-12N	SamacSys-Parts
IN		PINHD-1X2	1X02	pinhead
OUT		PINHD-1X2	1X02	pinhead
R1	1.3M	R-EU-R0805	R0805	rcl
R2	680k	R-EU-R0805	R0805	rcl
RMPPC	2M	R-EU-R0805	R0805	rcl
U1	CIVKA-JELF243	CIVKA-JELF243	CIVKA-JELF243A	civka

C.3 LTC-3129

Part	Value	Device	Package	Library
C1	22n	C-EUC0805	C0805	rcl
C2	22n	C-EUC0805	C0805	rcl
C3	4.7u	C-EUC0805	C0805	rcl
C4	2.2u	C-EUC0805	C0805	rcl
C5	47u	C-EUC0805	C0805	rcl
IN		PINHD-1X2	1X02	pinhead
L2	LQH3NPZ220MGRL	LQH3NPZ220MGRL	INDPM3030X100N	SamacSys-Parts
OUT		PINHD-1X2	1X02	pinhead
PS1	LTC3129IMSE-1#PBF	LTC3129IMSE-1#PBF	SOP50P490X110-17N	SamacSys-Parts
R1	1M	R-EU-R0805	R0805	rcl
R2	392k	R-EU-R0805	R0805	rcl
R3	1.3M	R-EU-R0805	R0805	rcl
R4	1M	R-EU-R0805	R0805	rcl
R5	2.4M	R-EU-R0805	R0805	rcl
R6	1M	R-EU-R0805	R0805	rcl

D Obsah přiloženého CD

Obsah přiloženého CD, společně s krátkými komentáři, je vypsán v následující adresářové struktuře.

```
/ ..... kořenový adresář přiloženého CD
├── ComService..... aplikace pro konfiguraci zařízení v C# - Visual studio 2019
├── dp_final_0..... kód pro mikrokontrolér - System Workbench for STM32
├── dps ..... dps v editoru Eagle 6.6.0
├── lambda_aws ..... lambda funkce, která se v cloudu Amazonu stará o ukládání
    naměřených dat do MySQL databáze - python
├── Spice... webová databázová aplikace pro vizualizaci naměřených dat v ASP.NET
    Core - Visual studio 2019
└── dp_raichl.pdf ..... Text práce
```